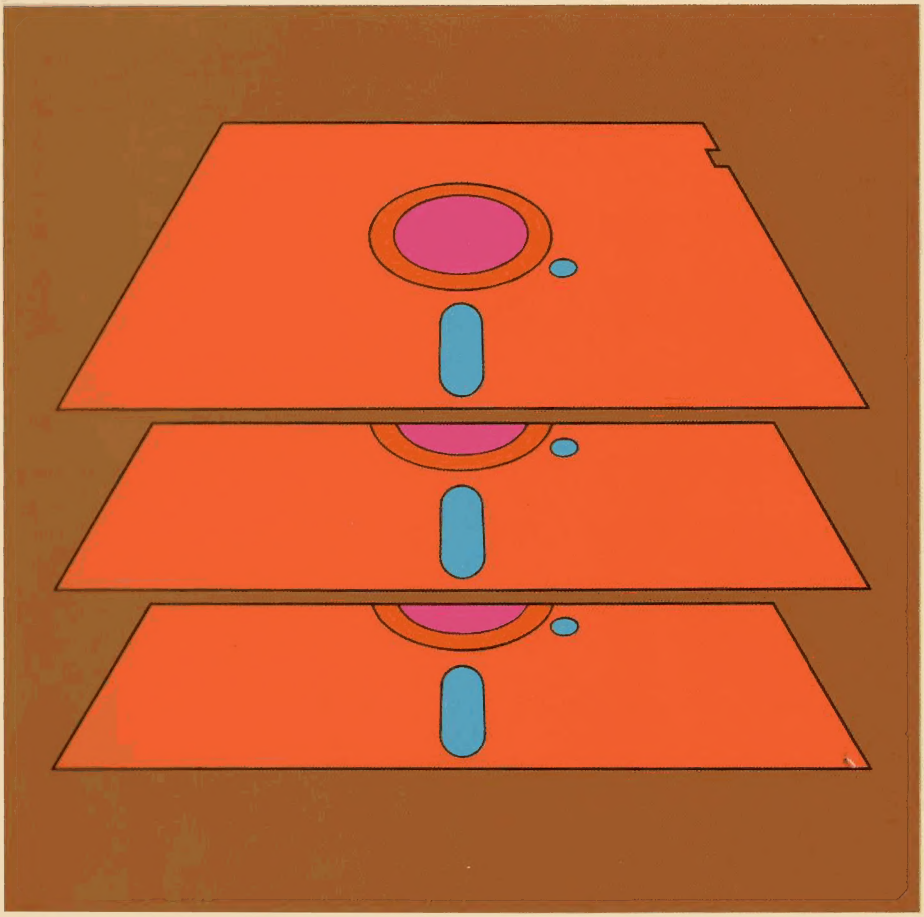
Apple II

Le Manuel DOS

Système d'exploitation sur disquette



AVERTISSEMENT

La Société Apple Computer Inc. se réserve le droit d'apporter des améliorations au produit décrit dans le présent manuel à tout moment et sans préavis.

REFUS DE TOUTES GARANTIES ET RESPONSABILITES

Apple Computer Inc. n'accorde aucune garantie, explicite ou implicite, en ce qui concerne le présent manuel ou le logiciel décrit dans ce manuel, sa qualité, ses performances, sa commercialisabilité ou son adéquation à un objet quelconque. Le logiciel d'Apple Computer Inc. est vendu ou licencié « en l'état ». La totalité du risque, pour sa qualité et ses performances, est supportée par l'acheteur. Si les programmes se révèlent défectueux à la suite de leur achat, l'acheteur (et non pas Apple Computer Inc., son distributeur ou son revendeur) supportera la totalité du coût de dépannage, de réparation ou de correction et des dommages indirects. En aucun cas, Apple Computer Inc. ne sera responsable des dommages directs ou indirects résultant d'un défaut quelconque du logiciel, même si Apple Computer Inc. a été avisé de l'existence de tels dommages.

Le présent manuel est protégé par Copyright. Tous les droits sont réservés. Le présent document ne peut, ni en totalité ni en partie, être copié, photocopié, reproduit, traduit ou transcrit sur un support électronique quelconque ou sous une forme lisible par une machine, sans le consentement préalable écrit d'Apple Computer Inc.

© 1980 par **Apple Computer Inc.**

Apple Computer International
7, rue de Chartres
92200 Neuilly-sur-Seine
FRANCE

« Apple » est une marque déposée de Apple Computer Inc.

N° de produit Apple A2LF0036. 030-F115-00.

Apple II

Le Manuel DOS

Système d'exploitation sur disquette

Table des matières

Préface

CHAPITRE 1
Le matériel

L'installation 2
La connexion du câble à la carte d'interface 2
L'insertion de la carte d'interface dans le micro-ordinateur 3
L'installation de plusieurs lecteurs de disquettes 5
L'emploi des lecteurs de disquettes et des supports disquettes 5
L'insertion et le retrait des disquettes dans le lecteur 6

CHAPITRE 2
La mise en route

Niveau de connaissances requis 10
Règles adoptées pour la représentation des touches spéciales du clavier 11
L'Apple du DOS 11
Si l'Apple du DOS ne fonctionne pas 12
L'initialisation de nouvelles disquettes 13
Le chargement et la sauvegarde avec le DOS 15
Le catalogue 16
Les noms de programmes et de fichiers 16
Les changements de noms 17
La suppression des programmes ou des fichiers 18
Comment faire si l'on passe accidentellement en mode moniteur 18

CHAPITRE 3
Les options – Applications

Les numéros de lecteurs, de ports et de volumes 22
La syntaxe 24
INIT 24
LOAD, RUN et SAVE 25
DELETE 27
Un scénario : appel du DOS, SAVE, RUN, CATALOG et DELETE 27
Changer de langage 28
L'utilisation du DOS à l'intérieur d'un programme 30

CHAPITRE 4
Les sécurités

La structure d'un système clé en main 34
LOCK et UNLOCK 35
VERIFY 35
La protection d'écriture sur disquette 36
Comment éviter les désastres 37
L'utilisation du programme COPY 38

CHAPITRE 5
Des informations de service

La mise au point : MON et NOMON 42
MAXFILES 43
TRACE 44
L'utilisation du programme UPDATE 44

CHAPITRE 6
Les fichiers à accès séquentiel

Les fichiers en mode Texte : introduction 48
Les fichiers à accès séquentiel : quelques exemples 49
L'ouverture et la fermeture des fichiers à accès séquentiel 57
L'écriture des fichiers séquentiels 58
La lecture des fichiers à accès séquentiel 62
APPEND et POSITION 64
La lecture et l'écriture partielles 67

CHAPITRE 7
Les procédures cataloguées

Le contrôle du système par un fichier en mode Texte : EXEC 70
La création d'une procédure cataloguée 71
La transformation d'un programme en un fichier en mode Texte 72
La conversion de routines Langage Machine en BASIC 73
La commande MAXFILES et le BASIC simple 74
La commande EXEC 74

CHAPITRE 8
Les fichiers à accès direct

Le fonctionnement des fichiers à accès direct 78
L'accès à un enregistrement déterminé 78
Plusieurs enregistrements 80
Un programme de démonstration de l'accès direct 81
L'écriture et la lecture des fichiers en accès direct 83

CHAPITRE 9
Le mode binaire

Les fichiers en Langage Machine 88
BSAVE 88
BLOAD 89
BRUN 89
La routine RWTS 89

CHAPITRE 10
Les entrées/sorties et le recouvrement de programmes

Le choix des unités d'entrée/sortie et divers 96
La commande CHAIN et le BASIC simple 100
La commande CHAIN et le BASIC étendu 101

ANNEXE A
Les types de fichier et leurs commandes du DOS

Les types de fichiers/programmes et les commandes correspondantes 104
Liste des types de fichiers/programmes et des commandes qui s'y rapportent 105

ANNEXE B
Les messages du DOS

Les codes d'erreur 108
Analyse des codes d'erreur 109

ANNEXE C
La mémorisation des informations sur disquette

Généralités sur la mémorisation 118
L'écriture d'un fichier à accès séquentiel en mode Texte 118
L'écriture d'un fichier à accès direct en mode Texte 119
Comment le DOS procède lors de l'écriture des fichiers en mode Texte 120
Le contenu des secteurs d'un fichier 121
La liste pistes/secteurs 121
Le catalogue du contenu d'une disquette 122

La table d'un volume 125
Le schéma d'occupation des pistes 126
Ordre d'affectation des pistes et des secteurs 128
Le chargement d'informations à partir de la disquette 129
La lecture d'un fichier à accès séquentiel 129
La lecture d'un fichier à accès direct 130

ANNEXE D
Schéma de la mémoire

Zones de mémoire utilisées par le DOS lors de l'appel 132
Zones de mémoire utilisées par le DOS et le BASIC (simple ou étendu) 133
Valeur de HIMEM fixée par l'appel du DOS 134

ANNEXE E
Points d'entrée du DOS

Points d'entrée du DOS 136
Schéma des circuits de l'interface DISK II 137
Schéma des circuits de la carte analogique du DISK II 138

ANNEXE F
Sommaire des commandes du DOS

Notation 140
Les noms de fichiers 142
Les commandes de service 143
Les commandes initiales 147
Les commandes pour les fichiers en mode Texte à accès séquentiel 149
Les commandes pour les fichiers en mode Texte à accès direct 152
Les commandes pour les fichiers en mode binaire 153

ANNEXE G
Sommaire des procédures du DOS

L'appel du DOS 156
L'initialisation d'une disquette 156
Comment faire si l'on passe accidentellement en mode Moniteur 156
L'utilisation du DOS à partir d'un programme 156
La création d'un système clé en main 157
La création et la consultation d'un fichier en mode Texte à accès séquentiel 157

Ajouter des enregistrements à un fichier séquentiel 158
Le contrôle du système par une procédure cataloguée 159
La création et la consultation d'un fichier en accès direct 159
La copie d'un fichier en mode Texte 160
Le recouvrement de programmes en BASIC étendu 160

ANNEXE H
Comment passer en DOS 16 secteurs

Comment installer les nouvelles PROMS 162
Système à plus de un lecteur de disquettes 163
L'utilisation du DOS avec la Carte Langage 164
La réalisation d'un système clé en main avec la carte langage 164

ANNEXE I
L'utilisation de la disquette BASICS

Le démarrage des disquettes avec BASICS 170

ANNEXE J
L'utilisation du programme FID

Le démarrage 172
Les noms de fichier et les caractères génériques 172
Les commandes 173
Que faire en cas d'erreur 177

ANNEXE K
L'utilisation du programme MUFFIN

Le démarrage 180
Les caractères génériques 183

Index

Index général 186
Index des programmes 189
Index des messages 189

Fiche cartonnée : Index des résumés de commandes
Index des résumés des procédures DOS

Préface

Ce Manuel a deux objectifs. Le premier est de vous familiariser avec l'utilisation du DOS (Disk Operating System) : les différents chapitres comportent des exemples qui complètent et illustrent les explications données sur les diverses commanes du DOS et leur fonctionnement. Le second objectif est que ce livre soit un Manuel de Référence.

Pour utiliser un lecteur de disquettes DISK II, il est nécessaire que vous disposiez au minimum de 16 K octets de mémoire RAM. Il est en fait fortement recommandé de disposer au moins de 32 K octets de RAM, étant donné qu'il resterait très peu de place pour le programme sur un 16 K. Par ailleurs, les programmes MUFFIN et FID, qui sont mémorisés sur la disquette Maître, nécessitent un minimum de 32 K de RAM pour être exécutés.

Les lecteurs de mini-disquettes souples APPLE DISK II vous permettent de mémoriser et d'accéder à des informations plus commodément et beaucoup plus rapidement qu'à partir de la cassette magnétique. Les informations sont mémorisées sur des petites « disquettes » qui ont un diamètre d'environ 5 pouces. Il s'agit d'un support en plastique, recouvert d'une pellicule magnétisée qui reste en permanence dans une pochette de protection carrée.

L'un des avantages les plus importants de l'utilisation de DISK II est le fait que l'information est enregistrée, et consultée, sous le nom qui lui a été affecté. Un programme qui gérerait des numéros de téléphone peut très bien se nommer « NUMÉROS DE TÉLÉPHONE ». Pour le mémoriser sur disquette, il suffit d'utiliser l'ordre de sauvegarde SAVE :

SAVE NUMÉROS DE TÉLÉPHONE

Ce programme peut être appelé en mémoire avec une commande tout aussi simple. Le nom « NUMÉROS DE TÉLÉPHONE » sous lequel ce programme est enregistré est un **nom de fichier.**

Le logiciel (ensemble de programme) qui prend en charge l'organisation des fichiers sur la mini-disquette, qui opère la sauvegarde et le chargement en mémoire, et une foule d'autres opérations utiles, est appelé le Disk Operating System (Système d'Exploitation sur Disquette), généralement abrégé en DOS. Certaines personnes prononcent « DOSSE », et d'autres prononcent « Dé oh esse ». Apprendre à utiliser le DOS consiste à se familiariser avec l'utilisation de quelques commandes propres au DOS, qui sont décrites dans ce Manuel. Ces commandes peuvent compléter soit le BASIC étendu APPELSOFT, soit le BASIC entier INTEGER, soit le langage machine.

A l'occasion dans ce manuel, vous verrez les symboles :

Ceci signifie que le paragraphe suivant décrit une caractéristique spéciale ou inhabituelle.

A lire soigneusement - le paragraphe suivant vous informe d'un risque de situation irréversible qui peut vous faire perdre votre programme.

CHAPITRE 1
Le matériel

2 L'installation
2 La connexion du câble à la carte d'interface
3 L'insertion de la carte d'interface dans le micro-ordinateur
5 L'installation de plusieurs lecteurs de disquettes
5 L'emploi des lecteurs de disquettes et des supports disquettes
6 L'insertion et le retrait des disquettes dans le lecteur

L'installation

La boîte DISK II comprend plusieurs éléments. Vous y trouverez notamment :

1) le lecteur de disquettes (boîtier principal);
2) la carte d'interface de communication entre le lecteur de disquettes et le micro-ordinateur, appelée aussi le *contrôleur*. Elle sera branchée dans le micro-ordinateur;
3) un câble ruban, attaché au lecteur de disquettes, et destiné à relier celui-ci au contrôleur;
4) une disquette *MASTER,* ou disquette Maître du système. Elle comporte le logiciel des disquettes;
5) une disquette « BASICS »;
6) ce manuel.

Étant donné que l'on peut brancher deux lecteurs de disquettes sur la même carte d'interface, il est possible d'acheter exclusivement un lecteur de disquettes dans le cas où il s'agit d'une adjonction. Le contenu de la boîte DISK II ne comportera alors pas tous les éléments décrits ci-dessus.

Il convient de conserver l'emballage : il sera utile dans le cas où vous auriez à transporter votre lecteur. Le renvoi de la carte de garantie à APPLE COMPUTER n'est en principe pas nécessaire si le DISK II a été acheté en Europe. En effet, dans ce cas, c'est le distributeur qui assure la garantie. Il est néanmoins recommandé de renvoyer la carte de garantie dans tous les cas à APPLE COMPUTER, afin de faire partie du fichier clients de APPLE, et de recevoir notamment la lettre d'information *CONTACT. CONTACT* permet d'être régulièrement informé des mises à jour, et des nouveaux produits.

*** ATTENTION ***

Avant de connecter ou de déconnecter
quoi que ce soit
sur le contrôleur ou sur le micro-ordinateur
couper l'alimentation électrique
CECI EST UNE RÈGLE ABSOLUE

La connexion du câble à la carte d'interface

Lors de l'utilisation, le lecteur de disquettes sera connecté à la carte d'interface, ou contrôleur, par l'intermédiaire du câble ruban. L'une des extrémités du câble est déjà attachée au lecteur de disquettes.

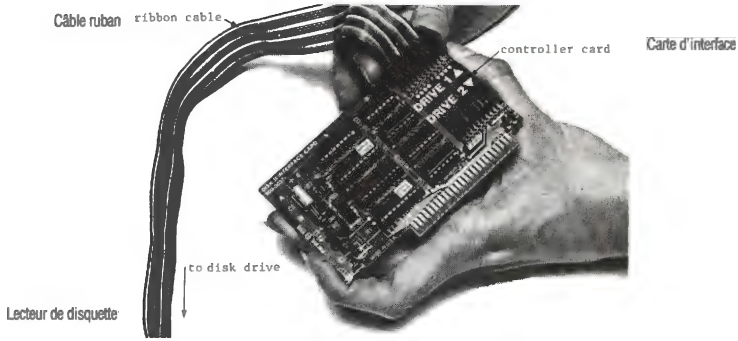
S'il s'agit du premier lecteur de disquettes, le câble doit être connecté sur les broches supérieures du contrôleur; elles sont repérées par le libellé « *DRIVE 1* ».

**** ATTENTION****

Si le câble n'est pas correctement connecté au contrôleur, le lecteur de disquettes peut être très sérieusement endommagé.

Tout d'abord, il convient de relier le câble au contrôleur *avant* d'insérer le contrôleur dans le micro-ordinateur.

Lorsque le câble est correctement connecté, le ruban ne doit pas être coincé entre le connecteur du ruban et le contrôleur, mais au contraire émerger du connecteur par la face opposée à celle qui est en contact avec le contrôleur. Chaque broche doit s'insérer aisément dans la loge correspondante sur le connecteur de câble. La connexion doit être faite avec douceur, et l'on doit éviter de toucher les contacts du contrôleur avec les doigts.



« La connexion du câble à la carte d'interface »

S'il s'agit du second lecteur de disquettes, le câble doit être connecté sur les broches inférieures du contrôleur; elles sont repérées par le libellé « *DRIVE 2* ». Il convient de prendre les mêmes précautions que pour la connexion du premier lecteur.

Si vous passez du DOS 3.2 au DOS 3.3, qui donne accès à 16 secteurs/piste, au lieu de 13, il sera nécessaire de changer deux composants PROM sur la carte d'interface. Reportez-vous à l'annexe H pour savoir comment procéder.

L'insertion de la carte d'interface dans le micro-ordinateur

Avant d'effectuer cette insertion, *il est indispensable de couper l'alimentation électrique du micro-ordinateur.* Si le micro-ordinateur était sous tension, l'insertion ou le retrait d'une carte d'interface, quelle qu'elle soit, risquerait d'endommager gravement à la fois la carte et le micro-ordinateur.

1. Mettre l'appareil hors tension.

2. Le couvercle du micro-ordinateur est fixé au moyen de deux boutons-pressions, non visibles, situés à l'arrière de l'appareil, à gauche et à droite. Retirer le couvercle en tirant d'un petit coup sec l'arrière du couvercle vers le haut. Dès que le couvercle est dégagé, le faire glisser horizontalement vers l'arrière, afin de le retirer sans le faire heurter de l'intérieur de l'appareil.

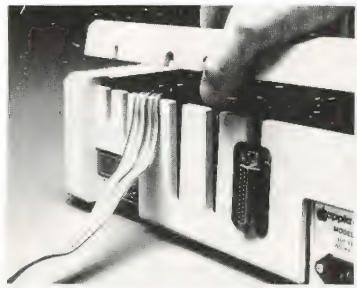
3. Le fond du micro-ordinateur est recouvert d'une carte de circuits électroniques. Cette carte comprend, dans sa partie la plus éloignée du clavier, huit ports d'entrée/sortie, numérotés de Ø à 7, à partir de la gauche. Les *ports d'entrée/sortie* peuvent être désignés sous le nom de *connecteurs d'interface.* En anglais, les ports d'entrée/sortie sont nommés des *slots.* La carte d'interface du lecteur de disquettes peut être placée dans un quelconque port d'entrée/sortie, excepté le Ø. APPLE place généralement l'interface de lecteur de disquettes dans le port 6, et la plupart des logiciels APPLE, ainsi que ce manuel, partent de l'hypothèse d'une connection du lecteur de disquettes dans le port 6.

4. Insérer les contacts du contrôleur dans le port 6, en évitant de les toucher avec vos doigts. Si vous les avez touchés par mégarde, il peut être préférable de les nettoyer en les frottant délicatement avec un coton imbibé d'alcool, afin d'assurer une parfaite transmission électrique. Lors de l'insertion de la carte d'interface

dans le port, vous ressentirez une certaine résistance. La carte doit être bien enfoncée : elle est ainsi maintenue en place fermement.

L'insertion de la carte d'interface

5. Placer le câble ruban afin qu'il repose à plat sur le bord du micro-ordinateur, entre deux fentes verticales. De cette manière, lorsque le couvercle du micro-ordinateur sera remis, il maintiendra le câble en place, évitant toute tension sur le contrôleur.



Le placement du câble

6. Remettre le couvercle du micro-ordinateur, en glissant la partie avant sous le boîtier, puis en appuyant à l'arrière, gauche et droit, pour fermer les boutons-pressions.

7. La carte d'interface du lecteur de disquettes est maintenant en place. Le lecteur de disquettes peut être placé à côté ou sur le micro-ordinateur.

L'installation de plusieurs lecteurs de disquettes

Chaque carte d'interface supporte deux lecteurs de disquettes, l'un connecté aux broches supérieures référencées DRIVE 1 (Lecteur 1), l'autre connecté aux broches inférieures, référencées DRIVE 2 (Lecteur 2)

La premmière carte d'interface peut être installée dans le port 6, la seconde dans le port 5, par exemple, et ainsi de suite.

Si le système comporte plusieurs lecteurs de disquettes, il est utile de les référencer par le numéro de port, suivi du numéro de lecteur, car c'est ainsi que le programme les distinguera.

L'emploi des lecteurs de disquettes et des supports disquettes

Le lecteur de disquettes DISK II, contrairement au micro-ordinateur, comporte une part importante de mécanique, avec un moteur et des pièces en mouvement. En conséquence, il est moins robuste que le micro-ordinateur. Un lecteur de disquettes doit être manipulé avec délicatesse. On doit éviter de le faire tomber, ou de faire choir des objets sur le lecteur; il ne doit pas être placé à proximité d'un récepteur de télévision, le champ magnétique de celui-ci pouvant endommager les propriétés magnétiques du lecteur. Une distance minimum de 60 cm du téléviseur doit être respectée.

Le support disquette utilisé sur le lecteur DISK II est une mini-disquette, de format standard (diamètre approximatif de 12 cm). La disquette, de matière plastique, est recouverte d'une pellicule magnétisable, semblable à la pellicule utilisée pour les bandes magnétiques. La disquette est protégée par une enveloppe extérieure, dans laquelle elle reste en permanence, même lorsqu'elle est en rotation. La disquette n'est jamais retirée de l'étui protecteur.

La mini-disquette est aussi appelé « *floppy* » mini-disquette ou mini-disquette souple, par opposition aux disques durs utilisés par d'autres ordinateurs, notamment les ordinateurs de l'informatique traditionnelle. Mais bien que la mini-disquette et son enveloppe protectrice soient souples, elles ne doivent pas être pliées ou tordues, même légèrement. L'enveloppe protectrice contient du lubrifiant et un traitement favorisant le nettoyage. Une propreté parfaite est nécessaire pour permettre la rotation de la disquette dans son étui. L'enveloppe doit aussi être manipulée avec précaution.

L'enveloppe protectrice comporte une fente par laquelle on aperçoit la disquette.

Une bonne manipulation du support conduit à saisir la disquette exclusivement par l'enveloppe protectrice : les doigts ne doivent jamais entrer en contact avec le support. Lorsqu'une disquette n'est pas utilisée, elle doit être rangée dans l'enveloppe de papier dans laquelle elle est livrée. L'enveloppe protectrice est, elle aussi, en effet traitée pour minimiser l'électricité statique, qui attire la poussière. Il est préférable de conserver les disquettes en position verticale. L'utilisation des boîtes en plastique dur conçues à cet effet est recommandée.

Les mini-disquettes permettent de mémoriser une grande quantité d'informations : une seule mini-disquette peut contenir jusqu'à 1 146 000 unités d'informations. Chaque unité d'information occupe donc une place infime de la surface du support. Une rayure invisible, ou une empreinte digitale à la surface peuvent être la cause d'erreurs. Il faut donc placer les disquettes en des endroits propres, à l'abri de la poussière et des cendres de cigarettes...

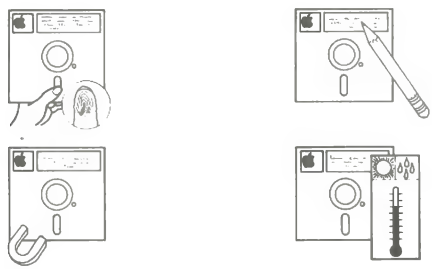
Les disquettes seront référencées, afin de pouvoir les distinguer. Lors de l'écriture d'une étiquette, il est préférable d'utiliser un feutre, et d'écrire sans exercer de pression. En fait, dans la mesure du possible, l'étiquette peut être établie séparément, puis apposée délicatement sur la disquette.

Les disquettes doivent être conservées à l'écart des champs magnétiques. Les moteurs électriques et les aimants en créent. Elles ne doivent pés être posées sur des appareils électroniques tels que les récepteurs téléviseurs. On peut les poser temporairement sur le micro-ordinateur ou sur le lecteur de disquettes.

Enfin, les disquettes sont sensibles à la température. Elles doivent être éloignées des sources de chaleur : soleil, radiateur. La température limite est de l'ordre de 50°C. Au-delà, les informations qui y sont mémorisées risquent d'être perdues. Lors du transport des disquettes, notamment en voiture, il faut prêter attention à ce qu'elles ne soient pas dans des conditions extrêmes de température. La première manifestation d'un excès de température est visible sur l'enveloppe protectrice qui se tord et se gondole.

Convenablement traitée, une disquette peut donner normalement environ 40 heures d'utilisation, ce qui est considérable par rapport aux quelques secondes nécessaires pour le chargement de la plupart des programmes.

Négligemment traitée, une disquette peut avoir une vie très brève, voire pratiquement aucune.

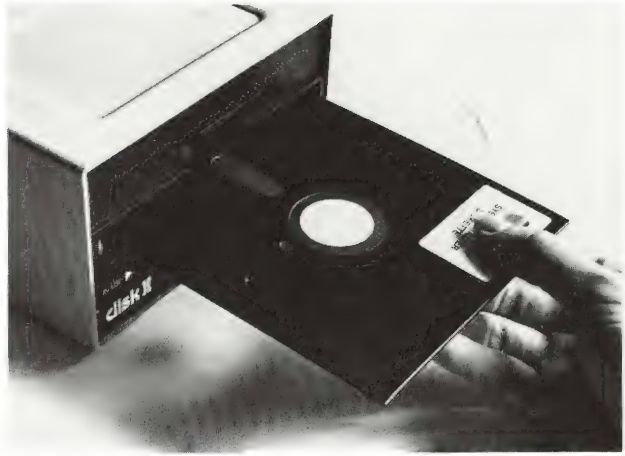


Fautes d'emploi

L'insertion et le retrait des disquettes dans le lecteur

L'utilisation du lecteur de disquettes est infiniment plus rapide que celle du lecteur de cassettes. Elle comporte néanmoins quelques contraintes d'utilisation.

La porte du lecteur s'ouvre lorsque l'on soulève la partie inférieure. La disquette, avec la *face comportant l'étiquette visible,* est introduite dans la fente. La partie de l'enveloppe protectrice comportant la découpe doit entrer en premier lieu, la partie comportant l'étiquette en dernier lieu. Une bonne règle à adopter, évitant en principe toute erreur, consiste à tenir la disquette avec le pouce droit sur l'étiquette.



L'insertion d'une disquette

Pousser doucement la disquette dans la fente, en prenant garde ne de pas la plier. Si la disquette est introduite trop brutalement, elle peut être définitivement détériorée.

La porte du lecteur de disquettes se ferme en l'abaissant.

Les deux languettes métalliques qui sont visibles dans la fente lorsque la porte est fermée se positionnent lors de l'introduction, sans toucher la disquette.

Pour retirer une disquette du lecteur, ouvrir la porte et tirer doucement l'enveloppe protectrice contenant le support.

Le fait de fermer la porte du lecteur abaisse la tête de lecture près du support. Le fait de l'ouvrir relève la tête de lecture. Si une disquette doit rester pendant un certain temps dans le lecteur sans être utilisée, il est préférable d'ouvrir la porte.

N'ouvrez jamais la porte d'un lecteur de disquettes lorsque le voyant rouge est allumé, indiquant qu'une opération est en cours. La conséquence presque certaine serait la perte des informations mémorisées et, de plus, le support serait probablement définitivement détérioré.

CHAPITRE 2
La mise en route

10 Niveau de connaisances requis
11 Règles adoptées pour la représentation des touches spéciales du clavier
11 L'appel du DOS
12 Si l'appel du DOS ne fonctionne pas
13 L'initialisation de nouvelles disquettes
15 Le chargement et la sauvegarde avec le DOS
16 Le catalogue
16 Les noms de programmes et de fichiers
17 Les changements de noms
18 La suppression des programmes ou des fichiers
18 Comment faire si l'on passe accidentellement en mode moniteur

Niveau de connaissances requis

Apprendre à utiliser le logiciel des disquettes, ou DOS — Disk Operating System — revient à compléter ses connaissances de BASIC par quelques instructions particulières, dont certaines ne sont d'ailleurs que des extensions d'instructions BASIC. C'est la raison pour laquelle la lecture de ce manuel implique une connaissance préalable du micro-ordinateur APPLE II et une maîtrise du langage BASIC.

Pour apprendre à utiliser le micro-ordinateur et pour maîtriser le BASIC simple, on peut consulter :

APPLE II Basic Programming Manuel — Apple Product #A2L0005X;

Pour maîtriser le BASIC étendu APPLESOFT, on peut consulter :

APPLESOFT II Basic Programming Reference Manual — Apple Product #A2L0006.

Il faut néanmoins, au préalable, maîtriser le BASIC simple.

Si vous êtes débutant, avant de poursuivre la lecture de ce manuel, vous devez apprendre le BASIC simple et, éventuellement, le BASIC étendu.

Des exemples de programmes illustrant l'utilisation du DOS sont donnés tout au long du manuel. La plupart de ces exemples sont écrits en BASIC APPLESOFT. Quelques-uns sont en BASIC simple. Dans certains, les modifications à apporter pour traduire du BASIC étendu en BASIC simple sont mentionnée. Dans d'autres cas, elles ne son pas signalées. Il convient de consulter à ce sujet l'annexe M du Manuel de BASIC étendu où toutes les différences d'écriture entre le BASIC étendu et le BASIC simple sont analysées.

Une expérience pratique complète avantageusement une lecture. Si vous en avez la possibilité. appliquez chacun des exemples donnés, au fur et à mesure, sur votre micro-ordinateur.

Mettre le micro-ordinateur en mode BASIC (BASIC simple ou BASIC étendu). Placer la disquette Maître (Master Disquette) dans le lecteur 1. La disquette Maître doit être référencée 004-0002-xx. Les deux derniers caractères sont remplacés par des X, car ils sont sans signification pour l'identification. Si deux lecteurs de disquettes sont connectés, utilisez le lecteur 1. Cette première partie du Manuel ne concerne que l'utilisation d'un seul lecteur, et présuppose qu'il est connecté sur le port # (numéro) 6.

Bien que vous ayez un lecteur de disquettes connecté, une disquette dans ce lecteur, et la porte fermée, vous pouvez constater que votre micro-ordinateur se comporte très exactement comme sans lecteur. Rien n'est modifié, tout se passe exactement comme si le lecteur de disquettes n'était pas là. En fait, le micro-ordinateur ignore la présence du lecteur de disquettes, tant qu'une commande spéciale ne lui a pas été donnée, l'informant de la présence du lecteur, de la disponibilité du logiciel des disquettes, et de ses nouvelles instructions.

Bien que les commandes du DOS ressemblent aux commandes BASIC, elles ne se conforment pas toujours aux mêmes règles. Par exemple, plusieurs commandes DOS ne peuvent pas être écrites sur une même ligne, séparées par une ponctuation. Un message "SYNTAX ERR" en résulterait.

Règles adoptées pour la représention des touches spéciales du clavier

* RETURN, signifie "frapper la touche RETURN". Il faut d'ailleurs frapper la touche RETURN après chaque instructions.

* RESET, signifie "frapper la touche RESET". Si votre micro-ordinateur est équipé d'une ROM autostart, cette opération aura pour effet l'affichage du caractère d'appel. Avec l'ancienne ROM, le fait de fraper la touche RESET ramène au mode Moniteur, qui se caractérise par l'affichage du caractère d'appel *.

* ESC, signifie "frapper la touche ESC". A l'origine, ESC signifiait "escape", c'est-à-dire évasion, mais son utilisation a été étendue.

* CTRL est quelque peu différent. Il convient d'enfoncer la touche CTRL (abréviation de *contrôle*), de la maintenir enfoncée, de frapper une autre touche, puis de relâcher les deux. Par exemple, *CTRL-C* signifie :
- enfoncer la touche CTRL,
- la maintenir enfoncée, et frapper la lettre C,
- relâcher.

*** ATTENTION ***

Les caractères frappés au clavier alors que la touche CTRL est enfoncée n'apparaissent pas à l'écran.

L'appel du DOS

Le fait d'ajouter au BASIC les commandes du logiciel des disquettes se nomme : l'appel du DOS (en anglais, *booting DOS*).

On peut appler le DOS de différentes manières à l'intérieur de chaque mode de fonctionnement. Dans les exemples ci-dessous, la lettre S remplace le numéro de port choisi. Le port standard est le #6. Après l'une quelconque des commandes qui suivent, il convient de frapper la touche RETURN.

Si l'on est en mode BASIC simple — le caractère d'appel est alors > — on peut utiliser l'une ou l'autre des deux commandes suivantes pour appeler le DOS :

PR#S	Exemples :	PR#6
ou IN#S		IN#6

Si l'on est en mode BASIC étendu — le caractère d'appel est alors] — on peut utiliser l'une ou l'autre des deux commandes suivantes pour appeler le DOS :

PR#S	Exemples :	PR#6
ou IN#S		IN#6

Si l'on est en mode Moniteur — le caractère d'appel est alors * — ont peut utiliser l'une des trois commandes suivantes pour appeler le DOS :

CS00G	Exemples :	C600G
ou S {CTRL} P		6 {CTRL} P

Dans la suite de ce manuel, lorsqu'il s'agira d'appeler le DOS, le détail de cette opération ne sera pas répété.

Les expressions "Boot the DOS", ou encore "Boot the disk", qui signifient "appeler le DOS" sont des expressions très employées par les utilisateurs de micro-ordinateurs. Le mot "boot" est l'abréviation de "bootstrap", et il vient d'une expression que l'on peut traduire par "démarrer à la force du poignet". Une étude étymologique plus approfondie serait déplacée. Sachez simplement que c'est une expression très utilisée aussi bien en informatique traditionnelle qu'en micro-informatique, et qu'elle ne signifie en aucun cas que l'on doive utiliser ses bottes pour donner un coup de pied au lecteur même si l'envie vous en vient à l'occasion!

Essayez maintenant d'appeler le DOS à partir de votre disquette Maître. Commencez par passer en mode BASIC, soit en BASIC simple, soit en BASIC étendu. Assurez-vous que la disquette est correctement introduite, puis tapez

```
PR#6
```

et frappez la touche RETURN, comme après chaque instruction (dernier rappel).

Dès que la touche RETURN a été enfoncée, la lampe témoin rouge du lecteur de disquettes s'allume et l'on entend des bruits insolites, miaulements et claquements. Ne vous alarmez pas. Après une dizaine de secondes, un message apparaîtra à l'écran. Le message devrait être en principe :

```
DOS VERSION 3.3          04/15/80
APPLE II STANDARD        SYSTEM MASTER
```

Si vous essayez d'utiliser le langage BASIC, vous constaterez que la plupart des commandes s'effectuent normalement, et en dehors du message qui vient de s'afficher, le micro-ordinateur semble n'avoir subi aucun changement. En fait, de nouvelles commandes ont été rendues accessibles, et certaines anciennes commandes offrent de nouvelles possibilités. Deux modifications, qui ne sont pas visibles, sont aussi intervenues :

– le pointeur d'adresse mémoire haute HIMEM a été modifié, pour tenir compte de la place occupée en mémoire centrale par le DOS;

– certaines capacités graphiques haute résolution ont peut-être été perdues (cette perte est fonction de la taille de la mémoire de votre micro-ordinateur).

Les versions du DOS qui utilisent 13 secteurs ne peuvent pas être appelées lorsque le système est prévu pour 16 secteurs (la disquette tourne toujours dans le vide, sans résultat). Pour utiliser une disquette du type 13 secteurs, il convient de la passer d'abord à 16 secteurs avec le programme MUFFIN. Une autre méthode consiste à utiliser la disquette BASICS pour appeler un DOS 13 secteurs sur un système à 16 secteurs. Reportez-vous à l'annexe I pour voir comment.

Si l'appel du DOS ne fonctionne pas

Relisez attentivement ce manuel. Cela résoud 90 % des problèmes de ce type.

Il est peu probable, mais il n'est pas impossible, que le connecteur à l'intérieur du lecteur de disquettes soit partiellement déconnecté. Si vous préférez ne pas tenter d'intervenir vous-même, votre distributeur sera ravi de vous dépanner.

Si, au contraire, vous préférez tenter vous-même l'intervention, procédez de la manière suivante :

– éteindre le micro-ordinateur;

– retirer la carte d'interface du micro-ordinateur;

– déconnecter le câble de la carte d'interface;

– dévisser légèrement les vis sous le lecteur, afin de permettre à la mécanique de coulisser dans le boîtier. Tirer vers l'avant;

– resserer les connecteurs, en les repoussant doucement sur la carte électronique;

– réassembler l'unité, et la reconnecter comme précédemment.

Si cette première intervention ne donne pas de résultat, ne tentez aucun autre réglage. Voyez votre distributeur.

L'initialisation de nouvelles disquettes

La disquette Maître du DOS est une disquette très spéciale : elle contient certains programmes qui vous permettront de copier une disquette en totalité, mettre à jour une disquette créée avec une version antérieure du DOS, et d'autres choses encore. Elle comporte aussi des programmes qui illustrent les différentes possibilités du DOS, et qui sont commentés dans ce manuel.

Retirez la disquette Maître du lecteur, et remplacez-la par une disquette vierge. Faisons une expérience. Tapez PR#6 (en mode BASIC) et observez ce qui se produit. La lampe rouge témoin s'allume, et le lecteur de disquettes après quelques claquements, fait entendre un bruit doux continu, et ne s'arrête plus. Pour l'arrêter, vous serez contraint d'appuyer sur la touche RESET (normalement ceci est tout-à-fait déconseillé, mais ces circonstances ne sont pas des circonstances normales).

En fait, le micro-ordinateur avait entamé une recherche infructueuse, et sans fin, sur une disquette vierge. Lorsqu'une disquette est fabriquée, elle ne contient aucune information, comme les cassettes vierges que vous pouvez acheter dans le commerce, à d'autres fins. Avant que la disquette ne puisse être utilement utilisée à l'aide d'un micro-ordinateur, il est nécessaire que des informations spéciales y soient enregistrées : la disquette doit être *initialisée*.

Si vous avez effectué les opérations avec votre micro-ordinateur, votre disquette vierge est dans le lecteur, et vous venez d'appuyer sur la touche RESET. Retirez la disquette vierge, remplacez-la par la disquette Maître, repassez en mode BASIC, et tapez PR#6. Vous avez de nouveau message affiché. Une fois de plus, nous venons d'ajouter les commandes du DOS au BASIC.

La commande *INIT* peut être utilisée pour initialiser les disquettes vierges, appelées en anglais "slave diskettes", c'est-à-dire disquettes esclaves, par oppositon à la disquette Maître. Nous les appellerons des disquettes secondaires.

La taille de la mémoire utilisateur du micro-ordinateur sur lequel une disquette est initialisée a son importance. Si une disquette est initialisée sur un micro-ordinateur de 32K, elle ne peut être utilisée que sur un 32K ou plus. Sur un micro-ordinateur de taille mémoire plus importante, seuls 32K seront utilisés. Après avoir initialisé une disquette vierge, vous pouvez utiliser le programme MASTER CREATE (voir chapitre 5), pour transformer votre disquette initialisée en une disquette Maître, dont le DOS est *relogeable* (dont l'adresse en mémoire est variable), de sorte que l'utilisation de la mémoire est optimalisée. Le programme MUFFIN vous permettra de transformer vos disquettes 13 secteurs en disquettes 16 secteurs, sans altérer le contenu.

La commande INIT implique l'utilisation d'un petit programme BASIC, appelé le programme de "salutations" puisqu'il accueille l'utilisateur : chaque fois que vous appellerez le DOS à partir de cette disquette, le programme de salutations sera automatiquement exécuté. Le programme de salutations est couramment appelé "BONJOUR", mais vous pourriez l'appeler "HELLO", ou "BUENOS DIAS", ou ce que vous voulez. Il

est préférable de se simplifier la vie en utilisant un nom standard, une fois adopté, pour les programmes de salutations de toutes vos nouvelles disquettes.

Voici comment procéder pour l'initialisation, le DOS ayant déjà été appelé :

– retirer la disquette Maître du lecteur, et la remplacer par une disquette vierge;

– taper NEW;

– taper un programme de salutations. Voici un exemple :

```
10 REM PROGRAMME DE SALUTATIONS 1
20 PRINT "DISQUETTE INITIALISEE SUR UN 32K"
30 PRINT "PAR AMY DOAKS, LE 8 AOUT 1982"
40 END
```

Vous devez, bien entendu, indiquer votre propre nom, la taille mémoire de votre micro-ordinateur, la date, et éventuellement d'autres informations qui vous permettront de connaître l'historique de la disquette.
Vous pouvez taper RUN pour voir si le programme de salutations s'exécute correctement.

– taper INIT BONJOUR

Lorsque vous frappez la touche RETURN, la lampe témoin s'allume, et l'on entend divers bruits, alors que la disquette se met en mouvement. L'opération d'initialisation dure environ deux minutes. Lorsqu'elle est terminée, la lampe témoin s'éteint et le caractère d'appel (> pour le BASIC simple ou] pour le BASIC étendu) réapparaît à l'écran.

– retirer la disquette, et la référencer par une étiquette. L'étiquette doit comporter, par exemple,

DISQUETTE INITIALISEE SUR UN 32K LE 8 AOUT 1982

afin qu'un rapide examen vous indique qu'elle n'est plus vierge.

Il convient de ranger la disquette Maître fournie par APPLE dans un endroit sûr, à l'abri de la chaleur, de la poussière, des champs magnétiques, etc. Et à un endroit dont vous vous souviendrez. Il faut en prendre soin, car la disquette Maître contient plusieurs programmes très utiles.

Une fois l'opération d'initialisation terminée, la disquette vierge est devenue une disquette secondaire. Après y avoir apposé une étiquette, remettez-la dans le lecteur, et appelez le DOS. Le message affiché à l'écran est celui que vous avez indiqué dans votre programme de salutations, par exemple :

```
DISQUETTE INITIALISEE SUR UN 32K
PAR AMY DOAKS, LE 8 AOUT 1982
```

Puisque la disquette permet l'appel du DOS, c'est que l'initialisation s'est correctement effectuée. Il convient maintenant d'utiliser la nouvelle disquette pour vos prochaines expériences. En effet, vous ne pourriez pas utiliser la disquette Maître pour certaines procédures, car elle comporte une *protection d'écriture*. Nous verrons au chapitre 4 à quoi sert la protection d'écriture, et comment la réaliser.

Si vous avez acheté plusieurs disquettes, ce serait une bonne idée d'en initialiser quelques-une maintenant.

Le chargement et la sauvegarde avec le DOS

Appelez le DOS à partir de la disquette que vous venez d'initialiser. Tapez NEW, pour vous assurer qu'il n'y a aucun programme en mémoire. Ceci aura pour effet d'effacer votre programme de salutations de la mémoire (programme qui est chargé, lu et exécuté lorsque vous appelez le DOS).

Tapez maintenant ce programme simple :

```
 5 REM PROGRAMME DE COMPTAGE
10 FOR I = 1 TO 10
20 PRINT I,
30 NEXT I
40 END
```

Exécutez-le une ou deux fois, pour vérifier qu'il se déroule comme prévu. En BASIC étendu, l'exécution donnera le résultat :

```
 1    2    3
 4    5    6
 7    8    9
10
```

Nous appellerons ce programe "DE UN A DIX", étant donné qu'il permet de compter de un à dix.

```
SAVE DE UN A DIX
```

Lorsque vous frappez la touche RETURN, la disquette se met en mouvement pendant quelques secondes, et le programme est sauvegardé.

Si vous aviez tapé simplement l'ordre SAVE, sans le faire suivre d'un nom, le programme aurait été sauvegardé comme d'habitude sur cassette (à condition que vous ayez préalablement connecté le lecteur de cassettes comme indiqué dans la brochure de BASIC).

Tapez LIST, puis RUN. Vous constatez que le programme est toujours dans la mémoire. Le fait d'utiliser le DOS pour sauvegarder un programme sur une disquette n'efface pas le programme de la mémoire.

Tapez NEW, puis LIST. Le programme a été effacé de la mémoire.

Pour être sûr que le programme n'est plus en mémoire, éteignez le micro-ordinateur. Vous pouvez aussi retirer la disquette du lecteur.

Mettre l'ordinateur de nouveau sous tension, remettre doucement la disquette en place, passer en mode BASIC, et appeler le DOS. Tapez NEW pour effacer le programme de salutations, puis LIST. Vous constatez qu'il n'y a aucun programme dans votre micro-ordinateur.

Tapez :

```
LOAD DE UN A DIX
```

La disquette se mettra en mouvement pendant quelques secondes, et le programme sera chargé en mémoire. Vous pouvez taper LIST, puis RUN et constater que le chargement s'est bien opéré. La sauvegarde et le

chargement à partir de la disquette sont des opérations simples. Les seules différences par rapport à l'utilisation de la cassette sont :

– que l'opération est nettement plus rapide;

– qu'un nom est donné au programme.

Le catalogue

Vous avez sauvegardé le programme DE UN A DIX sur votre disquette. En fait, vous aviez déjà sauvegardé un autre programme sur votre disquette. Pour connaître les programmes mémorisés sur une disquette donnée, il suffit de taper la commande :

```
CATALOG
```

et la liste de tous les programmes mémorisés s'affiche à l'écran. Si vos programmes ont été écrits en BASIC simple, votre affichage devrait être :

```
I 002 BONJOUR
I 002 DE UN A DIX
```

La lettre «I» dans la colonne de gauche signifie que les programmes sont écrits en BASIC simple, ou BASIC entier (Integer BASIC). La lettre A est inscrite devant les programmes écrits en BASIC étendu (APPLESOFT). Différents types de fichiers peuvent aussi être mémorisés sur disquette. Ce sujet est développé dans les chapitres 6 à 9.

Le nombre qui suit la lettre I représente la longueur du programme mémorisé. Dans ce cas particulier, 002 secteurs de la disquette sont utilisés. Chaque secteur permet l'enregistrement de 256 octets d'informations. L'enregistrement le plus court possible, un fichier vide en mode texte, par exemple (voir chapitre 6), nécessite 001 secteur, pour la mémorisation d'informations de service. En totalité, on peut mémoriser jusqu'à 496 secteurs de programmes ou de fichiers. Enfin, chaque entrée dans le catalogue comporte le nom du programme. Reportez-vous à l'annexe C pour de plus amples renseignements sur la manière dont l'information est mémorisée sur les disquettes.

Lorsque la taille d'un fichier excède 255 secteurs, la longueur affichée pour ce fichier repart de 000.

Il n'y a aucun moyen de repérer, en dehors de la lecture du catalogue, quel est le programme de salutations. C'est la raison pour laquelle il est utile d'adopter toujours le même nom pour le programme de salutations.

Parfois, il y a plus de programmes ou de fichiers dans un catalogue qu'il ne peut en être affiché sur une seule page d'écran. La commande CATALOG aura pour effet de faire apparaître les 18 premiers. Pour visualiser les suivants, il suffit de frapper une quelconque touche du clavier (sauf RESET, CTRL ou SHIFT).

Les noms de programmes et de fichiers

Leur longueur peut varier de 1 à 30 caractères. Au-delà de 30 caractères, le nom sera tronqué par le DOS. Le premier caractère est obligatoirement une lettre. Les caractères suivants peuvent être un quelconque

caractère du clavier (y compris l'espace), sauf la virgule.

Voici quelques noms autorisés :

ANTICONSTITUTIONNELLEMENT
DE UN A DIX
HIRES 24
STATISTIQUES : JEAN DUPONT

1 A 10 (le premier caractère n'est pas une lettre)
STATISTIQUES, JEAN DUPONT (contient une virgule)
DE LA MEME MANIERE DE LISSER UNE COURBE (sera tronqué après le 30e caractère)

Bien que dans le dernier exemple le nom sera tronqué après le 30e caractère lors de l'affichage du catalogue, on peut, si on le souhaite, taper le nom en entier lorsque l'on désire procéder au chargement (LOAD) ou à l'exécution (RUN).

Si la touche CTRL est actuellement (ou volontairement) frappée lorsque l'on tape un nom de programme ou de fichier, ce caractère n'apparaîtra pas à l'écran lors de l'affichage du catalogue. Par exemple, si vous tapez CTRL-T dans le nom AGATHA, à la place du T, le nom affiché par la commande CATALOG serait :

AGAHA

Pourtant, si vous essayez de charger le programme par la commande LOAD AGAHA, le micro-ordinateur répondrait :

FILE NOT FOUND

(c'est-à-dire fichier non trouvé dans le catalogue), bien que le nom tapé était *apparemment* identique. Il faut donc être attentif à ne pas introduire ce genre de caractère par erreur dans un nom. En contrepartie, il peut être intéressant de protéger un fichier contre des consultations dans les noms.

L'annexe F contient des informations sur la manière de repérer des caractères de contrôle contenus dans des noms.

Les changements de noms

Il est possible que vous souhaitiez changer le nom d'un programme ou d'un fichier. Supposez, par exemple, que vous vous lassiez de taper le nom "DE UN A DIX" et que vous préfériez le remplacer par le nom "COMPTEUR". Il suffit de taper :

RENAME DE UN A DIX, COMPTEUR

La disquette se mettra en route, et après quelques instants, le caractère d'appel réapparaîtra. Vous pouvez alors vérifier, à l'aide de la commande CATALOG, que l'opération s'est effectuée comme prévue.

La commande *RENAME* ne comprend pas de contrôle pour vérifier que le nouveau nom n'existe pas déjà dans le catalogue, pour un autre fichier, ou un autre programme. A la limite, vous pourriez, à l'aide de la commande RENAME, donner le même nom à tous les fichiers d'une même disquettes, créant ainsi une situation embarassante qu'il est préférable d'éviter.

La suppression des programmes ou des fichiers

Il est facile de supprimer un programme ou un fichier de la disquette.
Tapez CATALOG, pour voir les deux programmes que vous avez sur la disquette.
Tapez maintenant :

LOAD COMPTEUR

pour charger ce programme en mémoire.

Vous pouvez supprimer ce programme de la disquette par la commande :

DELETE COMPTEUR

et vérifier que cette commande a bien fonctionné en tapant CATALOG.

Il ne doit plus rester que le programme de salutations BONJOUR. Etant donné que le programme COMPTEUR est toujours en mémoire, voushpouvez l'écrire à nouveau sur la disquette avec la commande SAVE que vous connaissez déjà :

SAVE COMPTEUR

Une commande CATALOG vous permettra de vérifier que le programme est à nouveau sur la disquette.

Si vous essayez de supprimer un programme, ou un fichier, qui n'est pas sur la disquette, vous recevrez le message :

FILE NOT FOUND (fichier non trouvé).

Comment faire si l'on passe accidentellement en mode Moniteur

Supposons tout d'abord que vous ayez un programme en mémoire sans le DOS — un programme chargé directement à partir d'une cassette, par exemple, ou bien un programme tapé au clavier alors que vous n'aviez pas préalablement appelé le DOS. Si, par mégarde, vous frappez la touche RESET, vous pouvez généralement retrouver le BASIC (BASIC simple ou étendu), et votre programme intact, en tapant CTRL-C.

Si vous avez le DOS en mémoire alors que vous utilisez l'ancienne ROM Moniteur, et que vous passiez accidentellement en mode Moniteur (caractère d'appel *), vous pouvez généralement récupérer le BASIC, et votre programme en tapant 3DØG. Remarquez qu'il s'agit du chiffre Ø (zéro) entre les lettres D et G, et non la lettre O.

Si, alors que vous avez récupéré le BASIC (et votre programme que vous pouvez voir par LIST), vous n'avez pas récupéré le DOS, la seule maniére de vous en sortir est de passer par le lecteur de cassettes : sauvegarder le programme sur le lecteur de cassettes; appeler le DOS; charger le programme en mémoire à partir de la cassette; le sauvegarder sur la disquette. C'est la raison pour laquelle il est utile de conserver la disponibilité d'un lecteur de cassettes, même lorsque le micro-système possède un lecteur de disquettes.

STOP

Si la touche RESET est frappée alors que la lampe témoin du lecteur de disquettes est en fonctionnement, il est possible que les informations contenues sur la disquette soient détruites. Ce sera particulièrement le cas si c'est un ordre d'écriture sur disquette (SAVE, BSAVE, WRITE (qui s'exécutait au moment de l'interruption. Dans ce cas, vous ne pourrez probablement pas récupérer les programmes qui étaient mémorisés sur votre disquette. Vous pourrez toujours l'initialiser de nouveau, afin de pouvoir l'utiliser en tant que support, mais l'initialisation efface la totalité des programmes et des fichiers enregistrés sur une disquette.

Si la lampe témoin du lecteur de disquettes reste allumée pendant plusieurs minutes, et que vous n'entendiez pas les bruits habituels accompagnant les mouvements de la disquette, il est possible que votre système se soit "planté". Dans ce cas, le seul moyen d'en finir est de frapper la touche RESET, puis de redémarrer en appelant de nouveau le DOS.

Les informations contenues sur une disquette peuvent n'être que partiellement détériorées alors que l'on ne réussit pas à appeler le DOS à partir d'une telle disquette. On peut, dans certains cas, récupérer les programmes qui y sont enregistrés : on appelle le DOS à partir d'une autre disquette; puis on la remplace par la disquette défectueuse et on les sauvegarde sur une bonne disquette. Vous pouvez utiliser aussi le programme FID pour copier un à un des programmes ou des fichiers.

CHAPITRE 3
Les options-applications

22 Les numéros de lecteurs, de ports et de volumes
24 La syntaxe
24 INIT
25 LOAD, RUN et SAVE
27 DELETE
27 Un scénario : appel du DOS, SAVE, RUN, CATALOG et DELETE
28 Changer de langage
30 L'utilisation du DOS à l'intérieur d'un programme

Les numéros de lecteurs, de ports et de volumes

La plupart des commandes du DOS vous permettent de spécifier un certain nombre d'options, telles que le numéro du lecteur que vous utilisez, le numéro de port dans lequel le contrôleur du lecteur de disquettes est connecté, et le numéro de volume de la disquette.

L'option numéro de lecteur vous permet de travailler avec plus d'un lecteur de disquettes. Chaque carte d'interface, ou contrôleur, a la possibilité de contrôler soit un lecteur, soit deux lecteurs. Normalement, les instructions se réfèrent au lecteur 1. C'est le numéro de lecteur qui est *pris par défaut,* si aucun numéro de lecteur n'a été spécifié : si vous ne donnez aucune indication quant au lecteur que vous utilisez, c'est le lecteur 1 qui sera utilisé. Si vous utilisez le lecteur 2, il vous suffit de préciser D2 (drive 2) après le nom du fichier (ou après d'autres options disquettes) et séparé de celui-ci par une virgule. Par exemple, pour initialiser une disquette sur le lecteur 2, il suffit d'écrire

```
INIT BONJOUR,D2
```

Lorsque le lecteur 2 a été spécifié dans un ordre DOS, toutes les commandes DOS suivantes porteront sur le lecteur 2, jusqu'à ce que le lecteur 1 soit à nouveau spécifié dans un ordre DOS. Le lecteur 2 est maintenant devenu le lecteur par défaut. Après l'initialisation indiquée plus haut, à titre d'exemple, la commande CATALOG aurait pour effet l'affichage du catalogue de la disquette montée sur le lecteur 2. Si l'on désirait consulter le catalogue de la disquette montée sur le lecteur 1, il conviendrait d'écrire :

```
CATALOG,D1
```

Le lecteur par défaut deviendrait alors à nouveau le lecteur 1.

Si plus de deux lecteurs sont connectés au micro-ordinateur, des contrôleurs supplémentaires sont nécessaires. Ils seront placés dans des ports différents du premier (généralement placé en port #6). Vous pouvez spécifier un *numéro de port* dans un ordre DOS, en indiquant Sn, où S signifie Slot (port) et où n représente le numéro de port dans lequel est inséré le contrôleur de la disquette considérée. Sn sera séparé du nom de fichier ou des autres options par une virgule. par exemple, pour initialiser une disquette sur le lecteur 1 d'une paire de lecteurs dont le contrôleur est inséré dans le port 5, vous pouvez utiliser :

```
INIT BONJOUR,S5,D1
```

Le nom du fichier doit être indiqué en premier, mais l'ordre dans lequel les options sont données est indifférent.

Le numéro de port pris par défaut est celui qui a été utilisé lors de l'appel du DOS. Dès qu'un autre numéro de port a été spécifié, il devient le numéro par défaut, jusqu'à ce qu'un autre ordre DOS le modifie.

Si une commande du DOS comporte un numéro de port dans lequel aucun contrôleur de disquettes n'est inséré, le micro-ordinateur affichera un message d'erreur

```
I/O ERROR
```

signifiant Input/Output Error, soit, en français, Erreur d'Entrée/Sortie.

Après l'affichage de ce message, tout a l'air correct, mais en fait ne l'est pas. En effet, le DOS a conservé le

numéro de port erroné comme numéro par défaut, et considère que la disquette est toujours en mouvement. Même si la prochaine commande du DOS signifie un autre numéro de port, correct, cette fois, le DOS attend le signal selon lequel la disquette inexistante a terminé. Ce signal ne viendra jamais; il faut donc intervenir.

Si aucun programme n'est en mémoire, il suffit de rappeler le DOS.

Si, par contre, il convient de ne pas perdre un programme en mémoire :
- tapez CATALOG,Ss où s est le numéro correct de port;
- lorsque le système se plante, frapper la touche RESET;
- tapez 3DØG

et tout devrait rentrer dans l'ordre.

Le DOS doit être appelé à partir d'une disquette insérée dans le lecteur 1, et non dans le lecteur 2.

L'option *numéro de volume* peut être utilisé pour protéger les disquettes contre une utilisation non prévue. Supposez, par exemple, que vous ayez une application de tenue de stocks dans laquelle la situation de chaque mois est mémorisée sur une disquette comportant un numéro de volume unique. Dans une telle application, si l'on veut utiliser la disquette du mois de janvier, par exemple, il faut spécifier le bon numéro de volume. Sinon, il sera impossible d'écrire sur cette disquette : le DOS afficherait un message :

VOLUME MISMATCH (volume non conforme)

assurant ainsi un contrôle supplémentaire au niveau de l'exploitation.

On peut assigner un numéro de volume à une disquette lors de son initialisation en utilisant l'option Vn où n est le numéro de volume choisi.

Par exemple, pour initialiser une disquette, insérée dans le lecteur 2 d'une paire de lecteurs, dont le contrôleur est connecté au port 5, avec un programme d'appel nommé SALUT, et avec un numéro de volume 128, on écrirait :

INIT SALUT,D2,S5,V128

On ne peut modifier le numéro de volume d'une disquette qu'en la réinitialisant.

Le numéro de port, le numéro de lecteur et le numéro de volume peuvent apparaître dans un ordre quelconque dans une commande DOS. La commande précédente est équivalente à :

INIT SALUT,S5,V128,D2

et à

INIT SALUT,V128,S5,D2

et ainsi de suite.

Le numéro de volume d'une disquette doit être un nombre compris entre 1 et 254. Si aucun numéro de volume n'est spécifié lors de l'initialisation, c'est le numéro 254 qui sera pris par défaut.

La commande

INIT BONJOUR,VØ

n'occasionne aucun message, mais le numéro de volume assigné sera le numéro 254, c'est-à-dire le numéro de volume pris par défaut, puisque Ø n'est pas un numéro de volume autorisé.

Dans toutes les commandes du DOS, on peut spécifier un numéro de volume, si l'on désire vérifier la conformité du numéro annoncé avec celui qui est effectivement inscrit sur la disquette. Si aucun numéro de volume n'est spécifié, ou si la commande DOS comporte l'option V mais sans que celle-ci soit suivie d'un numéro, ou bien si le numéro spécifié est Ø, le DOS ignorera le numéro de volume de la disquette.

Par contre, si un numéro incorrect est spécifié, le DOS affichera :

VOLUME MISMATCH

Cette erreur ne peut pas se produire lorsque l'on demande le catalogue. Dans le cas où vous souhaiteriez connaître le numéro de volume d'une disquette, il suffit de taper CATALOG. Le numéro de volume est affiché en tête du catalogue.

Des informations complémentaires sont données à propos des options chaque fois qu'une nouvelle commande est présentée dans ce manuel.

Par ailleurs, ce type d'informations est résumé dans la nomenclature des commandes du DOS et sur la carte de référence qui l'accompagne. Vous trouverez ci-dessous le détail des conventions adoptées pour représenter symboliquement certaines données dans ces résumés.

La syntaxe

Nous désignerons par le mot syntaxe la structure d'une commande d'ordinateur, l'ordre et la forme correcte de ses différents éléments.

Une notation simple est utilisée pour décrire la syntaxe de chaque commande du DOS. Les éléments entre parenthèses sont optionnels. Les éléments optionnels d'une commande DOS peuvent être indiqués dans un ordre quelconque. Les lettres majuscules et les virgules doivent être reproduites telles quelles; les lettres miniscules représentent les mots fournis par l'utilisateur.

Les conventions suivantes sont adoptées pour la description des commandes DOS :

- f représente un nom de fichier
- d représente un numéro de lecteur — 1 ou 2
- s représente le numéro de port — 1 à 7
- v représente le numéro de volume — 1 à 254

Le numéro de volume d'une disquette ne peut pas être égal à Ø(zéro). Le fait de spécifier un numéro de volume Ø est une indication passe-partout qui indique au DOS qu'il convient d'ignorer le numéro de volume de la disquette.

Les autres abréviations utilisées dans ce manuel sont résumées au début de la nomenclature des commandes du DOS.

Toute constante numérique (le numéro de lecteur, le numéro de volume, etc...) dans une commande DOS peut être exprimée en notation hexadécimale, en faisant précéder les caractères hexadécimaux du signe $ dollar. Si vous n'êtes pas familier avec la notation hexadécimale, cela n'a aucune importance. Il n'est pas nécessaire de connaître la notation hexadécimale pour comprendre ce manuel.

INIT

La syntaxe pour la commande INIT est :

INIT f (,Vv) (,Ss) (,Dd)

où les parenthèses contiennent des options qui peuvent être omises.

L'exemple

INIT BONJOUR,V17,D2

peut être interprétée comme suit.

La commande INIT est en lettres majuscules, et doit être tapée exactement telle quelle. La lettre miniscule f, représentant le nom du fichier, est remplacée par le nom BONJOUR. Puis le numéro de volume optionnel est indiqué : en premier lieu on trouve une virgule, puis la lettre majuscule V. La lettre minuscule v a été remplacée dans cet exemple par le numéro 17. Les parenthèses entourant ",–Ss" montrent que l'on peut indiquer le numéro de port, ou omettre cette indication, dans la commande INIT. Dans cet exemple, le numéro n'est pas indiqué. Le DOS prendra donc le numéro de port par défaut. L'option lecteur (Drive) est, par contre, donnée : la virgule et la lettre majuscule D doivent être indiquées, suivies d'un chiffre (1 ou 2) qui remplace la lettre minuscule d = 2 dans cet exemple.

LOAD, RUN et SAVE

Le chargement, l'exécution et la sauvegarde des programmes sur disquettes sont semblables en tant qu'opérations à celles que l'on effectue sur cassettes, à la différence près que les programmes comportent un nom. Tout se passe au moins dix fois plus vite, et il n'est pas nécessaire d'actionner plusieurs touches pour enregistrer ou rembobiner... Tout se fait automatiquement. Il y a, de plus, d'autres possibilités et avantages qu'offre la disquette, tels que le catalogue des programmes, ou l'exécution automatique des programmes sans intervention de l'utilisateur. La sauvegarde des fichiers est aussi très facile (voir le chapitre 6 à ce sujet). Le programme FID décrit dans l'annexe J vous offre d'autres possibilités de gestion de fichiers.

Il est préférable de conserver la disponibilité d'un lecteur de cassettes. Il peut servir pour passer les programmes achetés dans le commerce, généralement vendus sur cassettes, sur vos disquettes. Il peut aussi vous servir pour archiver certains programmes, ou certains petits fichiers, bien que l'expérience montre que la disquette est un support encore plus fiable que la cassette.

Si vous avez un programme BASIC en mémoire, que vous désiriez le sauvegarder sur disquette en le nommant HENRI, il vous suffit de donner la commande :

SAVE HENRI

Si plus d'un lecteur est connecté au micro-ordinateur, HENRI aura été sauvegardé sur le lecteur à partir duquel le DOS a été appelé (à moins qu'un autre lecteur ait été spécifié dans une commande DOS postérieurement). On peut spécifier un numéro de port et un numéro de volume dans une commande SAVE, exactement comme dans une commande INIT. Par exemple, pour sauvegarder un programme nommé AGATHA sur la disquette de numéro de volume 214, insérée dans le lecteur 1 du contrôleur 2, on écrirait :

SAVE AGATHA,D1,S2,V214

Comme précédemment, l'ordre des informations optionnelles est indifférent. Si le numéro de volume n'avait pas été spécifié, le programme AGATHA aurait néanmoins été sauvegardé, mais le DOS n'aurait simplement pas contrôlé préalablement que le numéro de volume était conforme.

Rappelons que les noms de programmes, comme les noms de fichiers,
- ont une longueur maximum de 30 caractères;
- le premier caractère est une lettre;
- les caractères suivants sont n'importe quels caractères du clavier, sauf la virgule.

Pour charger un programme nommé AGATHA, il suffit de donner la commande

LOAD AGATHA

et si un programme de ce nom existe dans le catalogue, il sera chargé.

Si vous désirez exécuter le programme AGATHA après l'avoir chargé, vous pouvez bien entendu donner les commandes :

LOAD AGATHA

puis

RUN

Il est possible d'effectuer ces deux opérations en une seule commande :

RUN AGATHA

est une commande du DOS qui, dans un premier temps, charge le programme AGATHA en mémore et, dans un second temps, l'exécute.

La syntaxe pour les commandes SAVE, LOAD et RUN est indiquée ci-dessous :

SAVE f (,Ss) (,Dd) (,Vv)
LOAD f (,Ss) (,Dd) (,Vv)
RUN f (,Ss) (,Dd) (,Vv)

Exemples :

SAVE CHEZ NOUS,D1,S7
LOAD HAUT
RUN AMOK,S7

Si vous obtenez un message "SYNTAX ERROR" –lorsque vous tentez de sauvegarder un programme, c'est que :
– vous avez commis une faute de frappe;
ou le DOS n'a pas été appelé, ou n'est plus présent.

Essayez de retaper votre commande.
Si le DOS avait bien été préalablement appelé, essayez de le récupérer en tapant RESET, puis :

3D0G

Si le DOS n'a pas été appelé, *ne l'appelez pas maintenant!* L'appel du DOS efface tout le contenu de la mémoire. Il faut donc sauvegarder sur cassette le programme que vous avez en mémoire. Utilisant un lecteur de cassettes, tapez la commande :

SAVE

Appelez le DOS.
Utilisant le lecteur de cassettes, tapez la commande :

LOAD

pour recharger le programme en mémoire à partir de la cassette.
Vous pouvez maintenant le sauvegarder correctement sur la disquette.

Si une disquette est détérioriée (quelqu'un a peut-être tenté de l'agrafer dans un cahier!), ou si elle n'est pas initialisée, ou s'il n'y a pas de disquette dans le lecteur, ou encore si la porte du lecteur est ouverte, le message

I/O ERROR (I/O est l'abréviation de Input/Output, c'est-à-dire Entrée/Sortie) apparaîtra lors d'une tentative de sauvegarde (SAVE) ou de chargement (LOAD).

Vérifiez tous les éléments mentionnés, et corrigez. Il n'est pas nécessaire dans ce cas d'appeler à nouveau le DOS. Il suffit de recommencer.

Si vous utilisez la commande

LOAD STOCK

et que STOCK n'est pas un nom de programme mémorisé sur la disquette qui est dans le lecteur, le message :

FILE NOT FOUND

(fichier non trouvé) sera affiché.

Consultez le catalogue de la disquette, pour trouver le nom exact du programme. Tous les caractères et espaces doivent être reproduits, exactement tels qu'ils sont affichés dans le catalogue.

DELETE

Pour effacer un programme ou un fichier d'une disquette, il suffit d'utiliser la commande

DELETE

La syntaxe de la commande DELETE est la suivante :

DELETE f (,Ss) (,Dd) (,Vv)

Par exemple, la commande :

DELETE PERSONNEL,V34,D2,S1

efface un fichier nommé PERSONNEL d'une disquette dont le numéro de volume est 34, qui est insérée dans le lecteur 2 du contrôleur 1.

Lorsqu'un programme ou un fichier est effacé par la commande DELETE, les secteurs précédemment occupés sont libérés.

Un scénario : appel du DOS, SAVE, RUN, CATALOG et DELETE

Supposons que votre micro-ordinateur est en mode BASIC simple, et que votre disquette Maître est dans le lecteur. Voici un dialogue, tel qu'il pourrait apparaître sur l'écran de votre micro-ordinateur. Les informations tapées par vous sont soulignées, pour les différencier, bien qu'elles n'apparaissent pas soulignées sur votre écran.

<u>>PR#6</u> (aura pour effet d'effacer l'écran et de faire apparaître la page suivante)

```
DOS VERSION 3.3                  04/15/80
APPLE II STANDARD                DISQUETTE MAITRE
>CATALOG
DISK VOLUME 254                  *I 026 APPLEVISION
*A 006 HELLO                     *I 017 BIORYTHM
*I 018 ANIMALS                   *B 010 BOOT13
*I 003 APPLE PROMS               *A 006 BRIAN'S THEME
*I 006 APPLESOFT                 *B 003 CHAIN
```

```
*I 009 COLOR DEMO
*A 009 COLOR DEMOSOFT
*I 009 COPY
*B 003 COPY.OBJ0
*A 009 COPYA
*A 010 EXEC DEMO
*B 020 FID
*B 050 FPBASIC
*B 050 INTBASIC
*A 028 LITTLE BRICK OUT
*A 003 MAKE TEXT
*B 009 MASTER CREATE
*B 027 MUFFIN
*A 051 PHONE LIST
*A 010 RANDOM
*A 013 RENUMBER
*A 039 RENUMBER INSTRUCTIONS
*A 003 RETRIEVE TEXT

>NEW
>10 PRINT "JABBERWOCK"
>20 END
>SAVE DEMO
WRITE PROTECTED
```

A ce point, vous pouvez remplacer la disquette Maître par une disquette secondaire que vous avez préalablement initialisée, étant donné que, contrairement à la disquette Maître, elle n'est pas protégée contre l'écriture.

```
>CATALOG
DISK VOLUME 254
    I 002 HELLO
    I 002 COMPTEUR
>SAVE DEMO
>CATALOG
DISK VOLUME 254
    I 002 HELLO
    I 002 COMPTEUR
    I 002 DEMO
>NEW
>RUN
*** NO END ERR
>RUN DEMO
JABBERWOCK
>DELETE DEMO
>CATALOG
DISK VOLUME 254
    I 002 HELLO
    I 002 COMPTEUR
```

Changer de langage

Imaginons que vous soyez en mode BASIC simple, et que vous décidiez d'écrire un programme en BASIC étendu (APPLESOFT), ou d'utiliser votre micro-ordinateur comme une calculatrice, avec des valeurs décimales et une virgule flottante. Pour appeler le BASIC étendu sans détruire le DOS, il suffit de taper la commande

FP

(abréviation de Floating Point, c'est-à-dire de virgule flottante), et en quelques secondes, le BASIC étendu sera disponible.

Si le BASIC étendu n'était pas disponible (parce que la carte BASIC étendu n'est pas insérée, et que le BASIC étendu n'est pas non plus sur la disquette utilisée), le message

LANGUAGE NOT AVAILABLE

serait affiché.

La syntaxe pour la commande FP est :

FP (,Ss) (,Dd)

où le port et le lecteur, optionnels, permettent de spécifier la disquette contenant le BASIC étendu.

Si vous utilisez le BASIC étendu et le DOS, et que vous désiriez repasser en BASIC simple, il suffit de taper la commande :

INT

(abréviation de INTeger BASIC, c'est-à-dire BASIC entier, ou BASIC simple), pour retourner au BASIC simple en conservant la disponibilité du DOS.

La syntaxe pour cette commande est simplement :

INT

sans aucun paramètre. En effet, vous occasionneriez l'affichage d'un message d'erreur (SYNTAX ERR) si vous tentiez d'adjoindre des paramètres D ou S avec la commande INT.

Si vous tapez INT alors que vous êtes déjà en BASIC simple, tout programme qui serait en mémoire serait perdu.

De la même manière, si vous tapez FP alors que vous êtes déjà en BASIC étendu, tout programme qui serait en mémoire serait perdu.

Bien entendu, lorsque vous passez du BASIC simple au BASIC étendu, ou vice-versa, tout programme qui serait en mémoire serait perdu.

En plus de passer du BASIC simple au BASIC étendu, et inversément, il est possible que vous souhaitiez passer en mode Moniteur, tout en conservant le DOS. Que vous soyez en BASIC simple ou en BASIC étendu, vous pouvez passer en mode Moniteur par :

CALL - 151

Vous constaterez que le caractère d'appel * du mode Moniteur s'affiche alors. Pour retourner à votre BASIC d'origine, il suffit de taper 3D0G.

A partir du mode Moniteur, on peut aussi taper les commandes
- INT pour retourner au BASIC simple, ou
- FP pour retourner au BASIC étendu.

Dans les deux cas, le DOS sera disponible, mais tout programme qui se trouverait éventuellement en mémoire serait perdu.

Si, lors d'une commande FP, le message

PROGRAM TOO LARGE

s'affiche, tapez d'abord

INT

pour restaurer le système, puis

FP

Bien que votre disquette contienne un programme, écrit en BASIC simple et nommé APPLESOFT, ne tapez pas

RUN APPLESOFT

Si vous le faisiez, tout aurait l'air de se passer normalement, alors que ce ne serait pas le cas. Vous pourriez le constater en frappant la touche RESET, puis en essayant de retourner au BASIC étendu. Le DOS vous repasserait en BASIC simple, car le programme APPLESOFT est décrit en BASIC simple, et vous seriez dans l'embarras.

Pour recopier le programme APPLESOFT sur une autre disquette, il suffit de faire :

```
LOAD APPLESOFT
```

à partir de la disquette d'origine, puis de placer une disquette secondaire à la place de la disquette d'origine, et de faire :

```
SAVE APPLESOFT
```

L'utilisation du DOS à l'intérieur d'un programme

Il est très souvent utile de pouvoir exécuter une commande du DOS à partir d' un programme. Par exemple, vous pourriez souhaiter que votre programme de salutations affiche automatiquement le catalogue, ce qui pourrait être obtenu si ce programme contenait un ordre CATALOG. De nombreuses commandes du DOS peuvent être exécutées à partir d'un programme.

Elles le sont en les faisant précéder d'un ordre PRINT, suivi d'une chaîne alphanumérique débutant par le caractère CTRL-D, suivi de la commande du DOS.

Voci un programme écrit en BASIC étendu. Lors de son utilisation en tant que programme de salutations, les informations écrites en lignes 20 et 30 seront affichées, suivies de la liste des programmes ou des fichiers présents au catalogue.

```
5   REM PROGRAMME DE SALUTATIONS
10  D$ = CHR$(4) : REM CHR$(4) EST CTRL-D
20  PRINT "DISQUETTE SECONDAIRE CREE SUR UN 32K"
30  PRINT "PAR JEAN-CLAUDE MUSCAT LE 8 AOÛT 1980"
40  PRINT D$; "CATALOG"
50  END
```

Le programme ci-dessus illustre la manière dont il est recommandé de traiter le problème du CTRL-D en BASIC étendu : tout d'abord une constante alphanumérique nommée D$ et contenant uniquement le CTRL-D est définie; la fonction CHR$ est utilisée à cette fin. Cette constante peut ensuite être utilisée comme en ligne 40 :

```
40  PRINT D$; "CATALOG"
```

Remarquez le point virgule qui suit la constante alphanumérique D$ et les guillemets qui entourent la commande DOS. Le point virgule n'est pas obligatoire dans les ordres PRINT, en BASIC étendu. Si votre programme comporte beaucoup de commandes du DOS, pour gagner du temps, vous pouvez, en BASIC étendu, les simplifier et écrire :

```
40  PRINT D$ "CATALOG"
```

En BASIC étendu, on peut utiliser la fonction CHR$ pour définir CTRL-D. Mais il faut se rappeler que la valeur ASCII du CTRL-D est 4 en BASIC étendu. C'est pourquoi un commentaire (REMark) peut être utile. La fonction CHR$ n'est pas disponible en BASIC simple.

Que ce soit en BASIC simple ou en BASIC étendu, on peut définir le caractère CTRL-D en tapant le caractère à l'intérieur d'une constante alphanumérique :

```
D$ = "
```

puis on tape la lettre D, tout en maintenant la touche CTRL enfoncée; on relâche les deux touches et on tape un guillemet, ". Remarquez que le caractère CTRL-D n'apparaît pas à l'écran. En fait, la définition de la constante alphanumérique apparaîtra à l'écran sous la forme :

```
D$ = ""
```

Étant donné que les caractères de contrôle n'apparaissent pas à l'écran, il est utile d'insérer un commentaire (REMark), afin que le lecteur puisse savoir ce qui est effectivement contenu dans la constante alphanumérique.

Voici une version du programme précédent, mais cette fois en BASIC simple :

```
10  D$ = "" : REM IL Y A UN CTRL-D INVISIBLE ENTRE LES GUILLEMETS
20  PRINT "DISQUETTE SECONDAIRE CRÉÉE SUR UN 32K"
30  PRINT "PAR JEAN-CLAUDE MUSCAT LE 8 AOÛT 1980"
40  PRINT D$; "CATALOG"
50  END
```

Une seule commande DOS peut être utilisée par l'ordre PRINT. L'objet de l'ordre PRINT doit débuter par le CTRL-D, et se terminer par la commande DOS.

L'utilisation de la → (flèche) avant pour copier ou corriger un ordre BASIC contenant un caractère invisible aura pour effet d'effacer ce caractère.

Dans les commandes DOS exécutées dans un programme, le D$ doit être précédé d'un retour chariot (RETURN), faute de quoi il serait ignoré.

Par exemple, si vous exécutiez ce programme :

```
5   REM TESTCATALOG PROGRAM
10  D$ = "" : REM IL Y A UN CTRL-D
20  PRINT "TEST"
30  PRINT D$; "CATALOG"
40  END
```

TESTCATALOG serait affiché, et la commande DOS ne serait pas exécutée. En effet, le DOS n'a pas reconnu la commande, car le point virgule qui suit le PRINT "TEST" empêche le retour chariot (RETURN). Le D$ n'est donc pas précédé d'un retour chariot.

Pour corriger cette anomalie, il suffit de supprimer le point virgule en ligne 20. La commande DOS en ligne 30 sera alors reconnue, et le catalogue s'affichera.

Certaines commandes du DOS ne peuvent être utilisées qu'à l'intérieur d'un programme. Elles seront donc toujours précédées du caractère CTRL-D :

```
OPEN
APPEND
READ
WRITE
POSITION
```

D'autres commandes du DOS peuvent être utilisées soit en mode direct (exécution immédiate), soit à l'intérieur d'un programme, précédées du caractère CTRL-D :

CATALOG	BSAVE
SAVE	BLOAD
LOAD	BRUN
RUN	EXEC
DELETE	CLOSE
RENAME	CHAIN
LOCK et UNLOCK	PR#
MON et NOMON	IN#

La commande DOS MAXFILES peut être utilisée comme nous venons de le voir pour la commande CATALOG, dans un programme en BASIC étendu. Par contre, cette commande doit être utilisée d'une manière particulière dans un programme écrit en BASIC simple (voir à ce sujet la commande EXEC au chapitre 7).

La commande INIT ne peut être utilisée qu'en commande directe exclusivement, et jamais à partir d'un programme. De cruelles conséquences vous rappelleront cette règle si vous la violez.

CHAPITRE 4
Les sécurités

34 La structure d'un système clé en main
35 LOCK et UNLOCK
35 VERIFY
36 La protection d'écriture sur disquette
37 Comment éviter les désastres
38 L'utilisation du programme COPY

Nous avons déjà vu deux manières de protéger vos disquettes. Le numéro de volume, dont l'utilisation est développée au chapitre 3, vous permet de vous assurer que vos informations seront enregistrées sur la disquette adéquate. Les caractères de contrôle insérés dans les noms de programme, ou dans les noms de fichiers peuvent être aussi un moyen de protection efficace (voir le chapitre 2, ainsi que l'annexe F). Par exemple, si le catalogue affiche :

COMPTE BANCAIRE

mais qu'en fait, vos initiales sont insérées à un endroit particulier de ce nom, en tant que caractères de contrôle, il est peu probable que quelqu'un d'autre puisse accéder à ce fichier.

Dans ce chapitre, les différents moyens de protéger les programmes et fichiers sont développés. L'une ou plusieurs de ces techniques vous seront probablement utiles, à un moment ou à un autre. Voyons d'abord comment réaliser la structure d'un système clé en main.

La structure d'un système clé en main

Supposons qu'un médecin désire tenir la comptabilité de son cabinet sur un micro-ordinateur.

La situation idéale serait que son personnel n'ait qu'à :
– mettre le micro-ordinateur sous tension;
– frapper la touche RESET;
– taper 6 CTRL-P, puis frapper la touche RETURN,
pour qu'immédiatement le programme de comptabilité soit chargé, et prêt à fonctionner.

Étant donné que le programme de comptabilité communiquerait avec l'utilisateur en français (heureusement!), le personnel de ce médecin n'aurait pas besoin de connaître quoi que ce soit concernant le micro-ordinateur, ou le langage BASIC. Le micro-ordinateur serait devenu un outil pour la comptabilité, ses caractéristiques internes ayant perdu toute importance au bénéfice des caractéristiques d'emploi du programme de comptabilité. C'est le principe même du système « clé en main » : du point de vue de l'utilisateur, le micro-ordinateur est un appareil dédié à une ou plusieurs tâches spécifiques, et la mise en route est aussi simple que de tourner une clé dans une serrure. Dans ce cas particulier, « tourner la clé » revient à mettre l'appareil sous tension, et à frapper cinq touches du clavier. Il n'est pas nécessaire d'être un expert pour faire cela.

Vous pouvez utiliser le programme de salutations auquel vous avez donné un nom lorsque vous avez initialisé la disquette, pour faire de votre micro-ordinateur un système clé en main.

Supposons que vous souhaitiez qu'à chaque démarrage le programme COLOR DEMO (qui est fourni sur la disquette Maître) se mette automatiquement en marche. Voici comment procéder :
– initialiser une disquette, comme il est dit au chapitre 2;
– remplacer la disquette initialisée par la disquette Maître, et taper RUN COLOR DEMO;
– une fois que vous êtes satisfait de l'exécution de COLOR DEMO, taper CTRL-C pour l'interrompre, et retourner au BASIC;
– remettre la disquette initialisée dans le lecteur. Nous supposerons que vous ayez appelé votre programme de salutations « BONJOUR », lors de l'initialisation;
– le programme COLOR DEMO est maintenant en mémoire. Si vous tapez :

SAVE BONJOUR

le DOS effacera le programme d'origine nommé BONJOUR, et le remplacera par le programme COLOR DEMO, dorénavant appelé BONJOUR.

Votre programme de salutations est donc maintenant le programme COLOR DEMO.

Pour vérifier que tout marche correctement, appelez le DOS. Vous devez avoir la même exécution que précédemment, lors de l'essai de COLOR DEMO.

Vous venez de créer un système clé en main : chaque fois que le DOS est appelé à partir de cette disquette, le programme COLOR DEMO se chargera en mémoire et s'exécutera sans aucune autre intervention.

LOCK et UNLOCK

Il est parfois nécessaire de s'assurer qu'un programme ne sera pas effacé, par erreur, d'une disquette. La commande LOCK (verrouiller) le permet. Exemple :

LOCK NESS,D2

Le catalogue du contenu de cette disquette affichera maintenant un astérisque (*) devant le nom NESS.

Si l'on désire lever cette protection, la commande UNLOCK (déverrouiller) le permet. Exemple :

UNLOCK NESS

La syntaxe pour ces commandes est :

LOCK f (,Ss) (,Dd) (,Vv)
UNLOCK f (,Ss) (,Dd) (,Vv)

L'interprétation de cette notation est donnée au chapitre 3.

Si vous tentez d'effacer (DELETE) ou de changer le nom (RENAME) d'un programme ou d'un fichier qui est verrouillé (LOCK), le message

FILE LOCKED

sera affiché, et l'opération ne sera pas effectuée.

Ce message serait aussi affiché si vous tentiez de sauvegarder (SAVE) un programme sous le nom d'un programme existant dans le catalogue de la disquette, mais verrouillé, à condition que le programme que vous tenez de sauvegarder soit écrit dans le même langage que le programme verrouillé.

Par contre, si vous tentiez de sauvegarder un programme écrit dans un langage sous le même nom qu'un programme existant dans le catalogue de la disquette, mais écrit dans un autre langage, le message

FILE TYPE MISMATCH

serait affiché. Il conviendrait alors d'essayer à nouveau, en utilisant un nom différent.

VERIFY

Il se peut que, dans certains cas, une information ne soit pas correctement mémorisée sur une disquette. Cela peut arriver, par exemple, si la disquette est rayée, ou poussiéreuse. La commande VERIFY permet de contrôler le bon état de l'enregistrement des informations d'un fichier.

La syntaxe est celle habituelle des commandes du DOS :

VERIFY f (,Ss) (,Dd) (,Vv)

Exemples :

VERIFY SAM
VERIFY FINANCE-8,D2,V22

La commmande VERIFY contrôle la cohérence interne des informations enregistrées pour le fichier considéré. Si tout est cohérent, il n'y a pas de message, le caractère d'appel apparaît à la fin de l'opération :

> pour le BASIC simple,
] pour le BASIC étendu,
* pour le Moniteur.

Cependant, la commande VERIFY ne contrôle pas qu'un programme est correctement écrit. Si, par une manœuvre malencontreuse, un programme ou une partie d'un programme s'est trouvé "écrasé" en mémoire et que vous en fassiez la sauvegarde, le programme sera dans le même état sur la disquette, et la commande VERIFY ne détectera aucune anomalie.

Si la commande VERIFY détecte une incohérence, le message

I/O ERROR

(abréviation de Input/Output ERROR, c'est-à-dire erreur d'entrée/sortie) est affiché.

Si vous lancez une commande VERIFY pour un fichier qui n'est pas présent sur la disquette, le message

FILE NOT FOUND

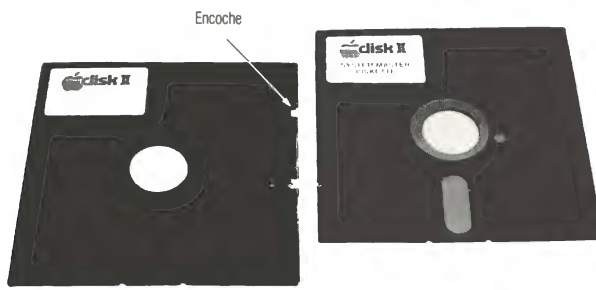
(fichier non trouvé) sera affiché.

La commande VERIFY peut être utilisée à partir du BASIC simple, du BASIC étendu ou du mode Moniteur. A partir de ces trois langages, on peut contrôler la cohérence interne de n'importe quel type de fichier, y compris des fichiers en mode texte (voir chapitres 6, 7 et 8) et les programmes écrits en langage machine (voir chapitre 9).

La protection d'écriture sur disquette

La commande LOCK vous permet de protéger un fichier. Il est possible que vous souhaitiez parfois protéger la totalité des programmes et des fichiers contenus sur une disquette.

Pour protéger une disquette contre l'écriture, il suffit de couvrir l'encoche située sur son bord gauche. De petites étiquettes autocollantes sont fournies à cet effet, avec les disquettes que vous achetez. Un bout de ruban adhésif fait aussi l'affaire. Remarquez que la disquette Maître n'a pas d'encoche : elle est en permanence protégée contre l'écriture.



Pour écrire à nouveau sur une disquette protégée contre l'écriture, il suffit de retirer l'étiquette recouvrant l'encoche.

Certains programmes ne peuvent pas être utilisés avec une disquette protégée contre l'écriture. Un exemple d'un tel programme est ANIMALS, l'un des programmes de démonstration enregistrés sur la disquette Maître.

Insérez votre disquette Maître dans votre lecteur, et appelez le DOS (si nécessaire). Tapez :

```
LOAD ANIMALS
```

ce qui chargera le programme en mémoire, puis

```
RUN
```

Le message

```
WRITE PROTECTED
STOPPED AT 1040
```

s'affichera.

Le programme ANIMALS ne peut pas s'exécuter avec une disquette protégée contre l'écriture, car ce programme mémorise des informations sur la disquette chaque fois que vous jouez.

Après avoir chargé le programme en mémoire (LOAD ANIMALS), il suffit de remplacer la disquette Maître par une disquette secondaire, non protégée contre l'écriture.
Vous pouvez alors taper

```
RUN
```

et constater que ANIMALS est un jeu qui se souvient de tout ce que vous lui apprenez, en mémorisant sur disquette les informations reçues. Quand vous aurez terminé le jeu, tapez

```
SAVE ANIMALS
```

de manière à disposer ultérieurement de ce jeu sur une disquette qui n'est pas protégée contre l'écriture.

Si vous tapez

```
CATALOG
```

vous constaterez que vous n'avez pas seulement sauvegardé le programme ANIMALS sur votre disquette, mais qu'un nouveau fichier, nommé ANIMALSFILE a été créé sur votre disquette par le programme ANIMALS.

Comment éviter les désastres

Les disquettes souples sont des supports robustes et sûrs, mais il est toujours possible de perdre ou de détruire les informations qui y sont mémorisées. Une disquette peut être rayée, détériorée par la chaleur, ou perdue. Si une disquette n'est pas protégée contre l'écriture, il se peut que l'on en efface les informations, par erreur. Par ailleurs, le support s'usera à la longue. Une durée de vie moyenne de 40 heures de fonctionnement est à en espérer.

Pour toutes ces raisons, il convient de conserver plus d'une copie de chaque programme. Nous dirons qu'il faut une possibilité de recouvrement (back up) pour tout programme de valeur.

Si vous êtes en train d'écrire, ou de modifier un programme, l'un des moyens est de conserver des copies des versions antérieures. Si la dernière version est perdue, vous pourrez toujours repartir de la version précédente, et ainsi ne pas perdre trop de temps de programmation. Pour adopter cette méthode sans risquer de se tromper, il convient d'ajouter un numéro au nom du programme qui représente le numéro de version, ce numéro progressant de un en un à chaque nouvelle sauvegarde.

Par exemple, supposez que vous écriviez un programme que vous nommez FINANCE. La première fois que vous sauvegardez ce programme, vous l'appellez FINANCE-1. La prochaine fois, après que vous l'aurez modifié, vous l'appellerez FINANCE-2, et ainsi de suite. Vous aurez ainsi une suite des différentes versions du programme FINANCE. Le numéro le plus élevé étant la version la plus récente.

Il est préférable de sauvegarder régulièrement un programme en cours de réalisation, avec un numéro de version croissant. Si vous le faites toutes les quinze ou vingt minutes, une interruption brutale de l'alimentation électrique, ou toute autre catastrophe, ne ruinera pas la totalité de votre travail. Vous pourrez, bien entendu continuer immédiatement votre programmation après avoir effectué une sauvegarde. Souvenez-vous simplement de faire progresser le numéro de version avant la prochaine commande SAVE. Si la disquette commence à être encombrée, il suffit de supprimer (DELETE) quelques-unes des versions les plus anciennes. Il est néanmoins utile d'en conserver plusieurs versions pour le cas où une quelconque catastrophe frapperait la dernière version! Parfois, même lorsque l'on dispose de la dernière version, il peut être utile de retrouver une version plus ancienne : les modifications ne sont pas toujours heureuses!

Une autre méthode de recouvrement consiste à conserver plusieurs copies d'un même programme sur différentes disquettes. Il y a deux moyens. Le premier peut fonctionner même si l'on ne dispose que d'un seul lecteur de disquettes : il suffit de sauvegarder le programme sur une disquette, puis de changer la disquette et de procéder à une autre sauvegarde, et ainsi de suite.

Le second moyen consiste à copier la totalité des informations d'une disquette sur une autre disquette. Ce moyen implique que vous disposiez de deux lecteurs de disquettes, et que vous utilisiez le programme COPY.

L'utilisation du programme COPY

Le programme COPY est fourni sur la disquette Maître. Il peut être utilisé pour sauvegarder la totalité du contenu d'une disquette sur un autre support. Si vous utilisez le DOS à partir du BASIC APPLESOFT, c'est le programme COPYA qu'il convient d'utiliser.

Dans le programme COPY, la disquette *à partir* de laquelle la copie sera effectuée est nommée « original ». La totalité du contenu de la disquette source sera copiée sur une disquette nommée « duplicata ». Il n'est pas nécessaire que la disquette réceptrice, duplicata, soit initialisée au préalable. De fait, toute information mémorisée sur cette disquette avant la copie sera totalement effacée.

Avant de copier la disquette d'origine, il est préférable de la protéger contre l'écriture. Ainsi, il est impossible de l'effacer par inadvertance, même si on se trompe de lecteur.

Le programme COPY prend par défaut le numéro de port et le numéro de lecteur en cours (donc le numéro de port et de lecteur à partir desquels vous avez commandé RUN COPY, ou RUN COPYA).

Pour utiliser le numéro de port et de lecteur par défaut pour la disquette d'origine, il suffit d'appuyer sur RETURN lorsque le programme demande un numéro. Si l'un des numéros par défaut ne convient pas, il suffit d'en donner un autre.

Lorsque le numéro de port et le numéro de lecteur ont été attribués pour la disquette d'origine, le programme COPY vous indique où mettre la disquette réceptrice "duplicata". Elle ira, par défaut, sur le lecteur restant ayant le même numéro de port que le lecteur de la disquette d'origine.

L'exemple suivant part de l'hypothèse que vous ne disposez que d'un seul lecteur de disquettes sur votre système.

1. Placer la disquette Maître dans le lecteur en cours. Taper :
   RUN COPY (BASIC simple),
   ou RUN COPYA (BASIC étendu).

   APPLE DISKETTE DUPLICATION PROGRAM

   ORIGINAL SLOT : DEFAULT = 6

2. Frapper la touche RETURN, pour indiquer que le numéro de port par défaut est accepté pour la disquette d'origine.

3. Le programme affiche ensuite :

   DRIVE : DEFAULT = 1

   frapper la touche RETURN, pour indiquer que le numéro de lecteur est accepté.

4. Lorsque le message

   DUPLICATE SLOT : DEFAULT = 6

   apparaît, frapper de nouveau la touche RETURN.

5. Lorsque le message donnant le lecteur par défaut pour la disquette dupliquée apparaîtra :

   DRIVE : DEFAULT = 2

   taper 1.

6. Le message

   --PRESS 'RETURN' KEY TO BEGIN COPY--
   (frapper la touche RETURN pour débuter la copie)

   s'affichera. Retirer la disquette Maître du lecteur, et la remplacer par la disquette originale.

7. C'est le programme qui vous donnera ensuite les instructions pour remplacer la disquette originale par la disquette dupliquée, et qui vous indiquera à quel moment il procède au formatage (INITialisation).
   Après le formatage de la disquette dupliquée, le programme vous demandera d'insérer à nouveau la disquette originale dans le lecteur, afin que ces informations y soient enregistrées. Cette double opération devra être répétée jusqu'à ce que la totalité de la disquette originale ait été copiée.

8. Lorsque la copie est réalisée, le message suivant s'affiche :

   DO YOU WISH TO MAKE ANOTHER COPY ?
   (désirez-vous effectuer une autre copie ?)

   Si vous répondez Y (yes-oui), la procédure de copie sera répétée, les mêmes numéros de lecteur et de port étant repris. Il convient de prêter attention à bien remettre l'original dans le lecteur lorsque le message

--PRESS 'RETURN' KEY TO BEGIN COPY--

apparaîtra à nouveau, et de la remplacer, au moment adéquat, par une disquette vierge.

Si vous ne désirez pas faire d'autre copie, répondez N (no-non). L'exécution quittera le programme. Le numéro de port et le numéro de lecteur par défaut qui seront adoptés par le DOS à la sortie sont ceux de la disquette dupliquée.

Si vous disposez de plus d'un lecteur, il convient de spécifier les numéros de port et de lecteur pour la disquette originale et pour la disquette dupliquée, et d'y insérer les supports disquettes avant de débuter la copie. Remarquez que si vous disposez de plus d'un lecteur, le système ne vous engagera pas à changer plusieurs fois les disquettes. Ceci n'est en fait nécessaire que dans le cas où vous ne disposez que d'un seul lecteur.

Si vous essayez de copier sur une disquette protégée contre l'écriture, le message

I/O ERROR
STOPPED AT... suivi d'un numéro de ligne du programme

s'affichera.

Tant que l'étiquette adhésive n'est pas retirée de la disquette dupliquée, aucune information ne peut y être enregistrée.

Un message "I/O ERROR" ou "UNABLE TO READ" (lecture impossible) peut aussi s'afficher si la porte du lecteur est ouverte, ou bien s'il n'y a pas de disquette dans le lecteur. Ces messages peuvent aussi être significatifs d'un quelconque problème concernant la disquette dans le lecteur. Attention : après l'un de ces messages, le numéro de lecteur par défaut a pu être modifié. Si vous exécutez à nouveau le programme COPY, il est possible qu'il soit nécessaire d'indiquer à nouveau le numéro de lecteur dans lequel la disquette Maître est insérée.

CHAPITRE 5
Des informations de service

42 La mise au point : MON et NOMON
43 MAXFILES
44 TRACE
44 L'utilisation du programme MASTER CREATE

La mise au point : MON ET NOMON

La mise au point des programmes est nommée, en américain "debugging", car les erreurs de programmes sont assimilées à des punaises (bugs) qu'il faut découvrir et retirer.

Toutes les commandes du DOS et toutes les informations transmises de l'ordinateur aux disquettes, et vice versa, ne sont normalement pas affichées à l'écran. Mais, lors de la mise au point des programmes, il peut être utile de visualiser ces informations à l'écran.

La commande MON (Monitor) le permet. La commande NOMON (No Monitor) permet de revenir à la situation normale.

Trois différents paramètres peuvent être utilisés avec cette commande :

C qui signifie Commande disquette, telle que OPEN, READ, etc.
I qui signifie INPUT (Entrée), les informations lues sur disquette,
O qui signifie OUTPUT (Sortie), les informations écrites sur disquette.

Ces paramètres ne sont utilisés qu'avec les commandes MON et NOMON. Normalement, le système fonctionne sous l'effet de NOMON C,I,O : aucun message du DOS n'apparaît à l'écran.

La syntaxe pour ces commandes est :

MON (C) (,I) (,O)
NOMON (C) (,I) (,O)

Au moins l'un des trois paramètres doit être présent avec les commandes MON et NOMON, faute de quoi elles seraient ignorées. Les paramètres peuvent être données dans n'importe quel ordre et, comme d'habitude, doivent être séparés par une virgule.

Il y a sept différentes manières selon lesquelles la commande MON peut être utilisée :

Commande	ce qui est affiché à l'écran,
MON C	les commandes DOS,
MON I	les informations lues sur disquette,
MON O	les informations écrites sur disquette,
MON I,O	les informations lues et écrites sur disquette,
MON C,I	les commandes DOS et les informations lues sur disquette,
MON C,O	les commandes DOS et les informations écrites sur disquette,
MON C,I,O	les commandes DOS et les informations lues et écrites sur disquette.

*** REMARQUES ***

Une commande MON reste effective jusqu'à ce qu'une commande NOMON, INT ou FP (BASIC étendu sur carte uniquement) soit donnée. L'appel du DOS ou le redémarrage par 3DØG restaurent aussi le NOMON C,I,O.

Un bon tuyau : vous pouvez donner une commande MON, et l'annuler ultérieurement sans affecter votre affichage d'écran. Même la commande NOMON ne s'affichera pas à l'écran. Supposons que vous exécutiez une commande MON, par exemple :

```
MON C,I,O
```

pour annuler cette commande sans affecter l'écran, insérez :

```
PRINT D$ : "NOMON C,I,O" : VTAB PEEK(37) : CALL -868
```

où D$ contient l'habituel CTRL-D ; PEEK(37) donne le numéro de ligne du dernier affichage. VTAB PEEK(37) repositionne donc le curseur. CALL -868 efface l'écran à partir de las position du curseur (équivalent de ESC-F en commande directe).

MAXFILES

Le DOS permet de traiter jusqu'à 16 fichiers actifs à un moment donné. Les fichiers gérés peuvent être de différentes natures : les programmes, les fichiers séquentiels en mode texte (voir chapitre 6), les fichiers à accès direct en mode texte (voir chapitre 8), et les fichiers en image mémoire binaire (voir chapitre 9).

La commande MAXFILES définit le nombre maximum de fichiers actifs autorisés. Lorsque le DOS est appelé, la commande MAXFILES 3 est automatiquement exécutée, ce qui définit trois fichiers au maximum, par défaut.

Le nombre maximum de fichiers autorisés est donc de 3, jusqu'à ce qu'une commande MAXFILES soit rencontrée.

La syntaxe de cette commande est :

MAXFILES n

où n est un nombre entier qui peut varier de 1 à 16. Si une valeur en dehors de ces limites est spécifiée, le BASIC (simple ou étendu) affiche un message SYNTAX ERROR. En mode Moniteur, un signal sonore sera la seule indication vous signalant l'erreur.

Pour chaque fichier annoncé, MAXFILES réserve 595 octets de la mémoire utilisateur comme mémoire tampon (en américain « buffer »). Cette place réservée pour chaque fichier en activité est nécessaire pour utiliser efficacement le système. En effet, il faut tenir compte du fait que la vitesse interne de la mémoire est considérablement plus élevée que le temps d'accès à la disquette qui implique le mouvement de pièces mécaniques, la tête de lecture devant se positionner au bon endroit sur la disquette.

Une mémoire tampon est donc utilisée dans laquelle transitent les informations allant vers, ou venant de la disquette.

Lors de la lecture d'informations sur disquette à partir d'un programme, le DOS alimente la mémoire par blocs de 256 octets, qui sont mémorisés dans la partie réservée aux entrées de la mémoire tampon. Puis le DOS envoie au programme une partie des 256 octets, correspondante à ce qui a été demandé.

Lors de l'écriture d'informations sur disquette à partir d'un programme, les caractères sont mémorisés dans la partie réservée aux sorties de la mémoire tampon, jusqu'à ce que 256 octets aient été accumulés. Ils sont alors physiquement écrits sur la disquette, en une seule opération.

Supposons que nous ayons MAXFILES 1, et qu'un fichier soit actif. Une tentative d'exécution d'une commande du DOS, telle que CATALOG, par exemple, nous donnerait le message :

NO BUFFERS AVAILABLE

(pas de mémoire tampon disponible).

Lorsque le DOS est appelé, le nombre de fichiers actifs est 3 par défaut, ce qui revient à réserver en mémoire 1785 octets pour les trois mémoires tampon correspondantes. Dans la plupart des cas, vous n'aurez pas besoin de plus de trois mémoires tampon. Si vous devez avoir plus de trois fichiers actifs, tapez :

MAXFILES n

(où n est le nombre de fichiers désirés), en commande directe, *avant* de charger et d'exécuter votre programme.

En commande directe, le fait de modifier le nombre de mémoires tampons efface les programmes écrits en BASIC simple, et le fait d'augmenter le nombre de mémoires tampons bouleverse les définitions de chaînes dans les programmes écrits en BASIC étendu. En effet, l'adresse de la mémoire haute (HIMEM) est baissée, sans que le programme ou les variables soient déplacés. Pour éviter ce problème, il suffit de fixer le nombre de maxfiles *avant* de charger et d'exécuter un programme.

Si la commande MAXFILES est utilisée à l'intérieur d'un programme, elle change les pointeurs, et les instructions GOTO et GOSUB peuvent ne plus fonctionner correctement. S'il est indispensable que vous ayez une commande MAXFILES à l'intérieur d'un programme écrit en BASIC étendu, il faut que la commande MAXFILES soit la première instruction du programme (il faut notamment qu'elle intervienne avant une quelconque déclaration de variable alphanumérique). Par exemple :

```
10 PRINT CHR$(4) ;"MAXFILES 5"
```

Pour utiliser la commande MAXFILES à l'intérieur d'un programme écrit en BASIC simple, il faut créer un fichier EXEC (voir chapitre 6 à la fin).

TRACE

L'instruction BASIC TRACE est un intéressant outil d'aide à la mise au point. Mais lorsque l'on utilise TRACE, les commandes DOS ne fonctionnent pas, car TRACE affiche les numéros de ligne sans retour chariot avant la commande DOS.

Il existe une solution très partielle au problème. On peut insérer un retour chariot dans la constante alphanumérique D$. Par exemple, en BASIC étendu :

```
10 D$ = CHR$(13) + CHR$(4)
```

Dans ce cas, la plupart des commandes DOS fonctionneront correctement de cette manière, avec TRACE, sous réserve de la remarque ci-dessous :

Si le mode TRACE est en fonction, et que le DOS tente une lecture de la disquette, l'exécution se bloquera sur l'ordre INPUT, l'ordinateur attendant une entrée à partir du clavier. Cette caractéristique limite l'intérêt de la commande TRACE utilisée sous DOS.

L'utilisation du programme MASTER CREATE

Comme nous l'avons vu au chapitre 2, la commande INIT est utilisée pour créer des disquettes secondaires. Nous allons voir maintenant comment créer des disquettes Maîtres. La différence entre une disquette secondaire et une disquette Maître n'est pas évidente. Leur apparence extérieure est presque identique.

La disquette Maître qui accompagne le lecteur de disquettes contient un programme nommé MASTER CREATE qui peut être exécuté sur un système 16K octets minimum. Ses fonctions sont les suivantes :

– transforme une disquette secondaire (dont le DOS est fonction de la taille de la mémoire utilisateur disponible lors de l'initialisation) en une disquette Maître (dont le DOS est « relogeable », utilisant ainsi avec un maximum d'efficacité la mémoire disponible, quelle qu'en soit la taille);

– donne un nouveau nom au programme de salutations. Le DOS tentera d'exécuter ce programme chaque fois qu'il sera appelé à partir de cette disquette.

Le programme MASTER CREATE doit être utilisé avec une disquette qui a été préalablement initialisée. Il ne fonctionne pas avec une disquette protégée contre l'écriture.

Voici un exemple qui montre comment mettre à jour la disquette initialisée au chapitre 2 (celle qui comporte le programme DE UN A DIX), pour transformer la disquette secondaire créée par INIT en une disquette Maître. Pour plus de commodité, nous appellerons la disquette secondaire la disquette 1 dans l'exposé de cet exemple.

Avant d'utiliser MASTER CREATE, effectuez les opérations suivantes :

1. Insérez la disquette que vous désirez mettre àjour — disquette 1, dans cet exemple — dans le lecteur, et tapez :

   RUN BONJOUR

   BONJOUR étant le nom du programme de salutations de la disquette 1. Le message affiché par le programme de salutations devrait comporter l'indication de la version du DOS utilisée pour initialiser la disquette, et son état : disquette secondaire ou disquette Maître.

2. Apportez les modifications adéquates au programme de salutations, afin que désormais ce programme affiche, lors de son exécution :

   DISQUETTE MAITRE

   Il convient de sauvegarder cette nouvelle version du programe de salutations, et de modifier l'étiquette collée sur la disquette.

3. Remarquez le nom du programme de salutations. Si vous souhaitez le conserver, il vous suffira de donner ce nom, lors de l'utilisation du programme MASTER CREATE. Par contre, si vous souhaitez depuis longtemps modifier ce nom, profitez-en, c'est le moment : à l'aide de la commande RENAME, donnez-lui un nouveau nom. Vous donnerez ultérieurement le nouveau nom au programme MASTER CREATE.

   Pour utiliser MASTER CREATE, procédez comme suit :

4. Insérez la disquette Maître fournie avec le système dans le lecteur, et appelez le DOS. Que vous soyez en BASIC simple ou en BASIC étendu, tapez :

   BRUN MASTER CREATE

5. Le message suivant sera affiché à l'écran :

   DOS 3.3 MASTER CREATE UTILITY
   COPYRIGHT 1980 BY APPLE COMPUTER INC.
   REMEMBER THAT MASTER DOES NOT CREATE THE 'GREETING' PROGRAM,
   PLACE THE DISKETTE TO BE 'MASTERED' IN THE DISK DRIVE

6. Puis le programme vous demandera le nom du programme de salutations par le message :

   PLEASE INSERT THE 'GREETING' PROGRAM'S FILE NAME

   (donnez s'il vous plaît le nom du programme de salutations). Nous supposerons que lorsque vous avez sauvegardé votre programme de salutations, après modifications, vous avez conservé le nom « BONJOUR ». Tapez :

   BONJOUR

   (à moins que vous ne souhaitiez que le DOS exécute un programme d'un autre nom chaque fois qu'il est appelé). Lorsque vous frapperez la touche RETURN pour envoyer le nom du programme, le message suivant s'affichera :

   REMEMBER THAT UPDATE DOES NOT CREATE THE "GREETING" PROGRAM,
   OR PLACE IT IN THE DISK DIRECTORY
   THIS IS THE FILE NAME THAT WILL BE PLACED WITHIN THE IMAGE
   BONJOUR
   PLACE THE DISKETTE TO BE 'UPDATED' IN THE DISK DRIVE
   PRESS (RETURN) WHEN READY
   NOTE : IF YOU WANT A DIFFERENT FILE NAME, PRESS (ESC)

   ce qui signifie :

7. Souvenez-vous que le programme « MASTER » ne crée pas le programme de salutations et ne l'insère pas dans le catalogue.
   Le nom qui sera enregistré est

   BONJOUR

   Placez la disquette à mettre à niveau dans le lecteur.
   Frappez la touche RETURN lorsque tout est prêt.
   Remarque : si vous désirez un nom de programme différent, frappez la touche ESC.
   Il suffit de suivre les indications. Remplacer la disquette Maître du système par la disquette 1. Frappez la touche RETURN, pour commencer la mise à jour; le programme vous informera lorsque l'opération sera terminée.

8. Après avoir utilisé le programme MASTER CREATE, il faut toujours appeler à nouveau le DOS avant de passer un nouveau programme.

*** REMARQUE ***

Le nom du programme de salutations que vous donnez au programme MASTER CREATE n'est pas automatiquement inséré dans le catalogue par ce programme. Cette indication sert uniquement à donner le nom du programme enregistré sur la disquette que le DOS doit exécuter chaque fois qu'il est appelé. C'est à vous de vous assurer que le catalogue de la disquette contient effectivement un programme portant le nom que vous avez indiqué au programme MASTER CREATE.

Si ce n'était pas le cas, le message

FILE NOT FOUND

(fichier non trouvé) serait affiché à chaque fois que vous appeleriez le DOS à partir de cette disquette.

*** RAPPEL ***

Il est indispensable de se rappeler des noms des programmes de salutations pour chaque disquette. Le plus simple est d'utiliser un nom commun à toutes les disquettes.

CHAPITRE 6
Les fichiers à accès séquentiel

48 Les fichiers en mode texte : introduction
49 Les fichiers à accès séquentiel : quelques exemples
57 L'ouverture et la fermeture des fichiers à accès séquentiel
58 L'écriture des fichiers à accès séquentiel
62 La lecture des fichiers séquentiel
64 Apprend et position
67 La lecture et l'écriture partielles

Les fichiers en mode texte : introduction

Vous souhaitez parfois utiliser la disquette pour mémoriser des informations autres que des programmes. Vous pourriez, par exemple, souhaiter conserver des copies de correspondances, une liste de mots utilisés dans un jeu, des résultats intermédiaires de calculs, ou une liste pour faire de l'adressage. Un fichier en mode texte vous le permettra notamment. La lettre T est placée devant un fichier en mode texte, lors de l'affichage du catalogue.

Les fichiers en mode texte sont créés et consultés par l'intermédiaire de commandes du DOS utilisées dans un programme BASIC, simple ou étendu.

Un fichier en mode texte peut être créé par un programme écrit dans un langage donné, et consulté dans un autre programme, écrit dans un langage différent.

La plupart des exemples de ce manuel sont en BASIC étendu. Si vous désirez les convertir en BASIC simple, rappelez-vous qu'en BASIC simple on ne peut pas définir des tableaux alphanumériques, et que les chaînes de caractères doivent être dimensionnées. En BASIC simple, une instruction telle que :

```
INPUT A$, B$, C$
```

nécessite un retour chariot RETURN, et non la virgule, entre chaque réponse, au moment de l'exécution.

Il n'est pas dit dans ce manuel comment convertir chacun des exemples en BASIC simple. Voir l'annexe M de la brochure de BASIC étendu à ce sujet (APPLESOFT II BASIC Programming Reference Manual). On peut aussi voir le chapitre 7 de ce manuel pour la conversion, après l'écriture, d'un programme d'un BASIC APPLE à un autre BASIC APPLE.

Les commandes LOAD et RUN doivent porter sur un programme BASIC, et les commandes BLOAD et BRUN doivent porter sur un programme en langage machine (image mémoire binaire), et non sur un fichier en mode texte. Il faut écrire des programmes pour enregistrer les informations et pouvoir les consulter dans un fichier en mode texte, utilisant les commandes du DOS qui sont développées dans ce chapitre :

```
OPEN
CLOSE
READ
WRITE
APPEND
POSITION
EXEC
```

Les commandes OPEN, READ, WRITE, APPEND et POSITION ne peuvent pas être utilisées en commande directe. Si vous le tentiez, le message :

```
NOT DIRECT COMMAND
```

serait affiché.

Ces commandes doivent être utilisées en différé, c'est-à-dire depuis un programme.

Les commandes CLOSE et EXEC peuvent être utilisées en commande directe.

En plus des commandes listées ci-dessus, les commandes DOS :

LOCK et UNLOCK
DELETE
RENAME

MON et NOMON
VERIFY
CATALOG

fonctionnent avec les fichiers en mode texte, de la même manière qu'avec les programmes.

Il y a deux types de fichiers en mode texte : les fichiers à accès séquentiel et les fichiers à accès direct. Ces deux types de fichiers mémorisent des chaînes de caractères représentés sous leur forme de valeur ASCII.

Dans le schéma ci-dessous, le caractère ↕ représente le RETURN.

REPRESENTATION D'UN FICHIER EN MODE TEXTE A ACCES SEQUENTIEL

Caractères :	7	↕	A	T	↕	O	N	E	↕	B	L	O	W	↕								
Valeur ASCII :	55	13	65	84	13	79	78	69	13	66	76	79	87	13	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	ØØ	
Numéro d'octet du fichier :	Ø	1	2	3	4	5	6	7	8	9	1Ø	11	12	13	14	15	16	17	18	19	2Ø	
Numéro de rubrique dans l'enregistrement	Ø		1			2					3											

REPRESENTATION D'UN FICHIER EN MODE TEXTE, A ACCES DIRECT :
(exemple : longuer de l'enregistrement 5, une seule rubrique par enregistrement).

Caractères :	7	↕				A	T	↕			O	N	E	↕		B	L	O	W	↕		
Val. ASCII :	55	13	ØØ	ØØ	ØØ	65	84	13	ØØ	ØØ	79	78	69	13	ØØ	66	76	79	87	13	ØØ	
Nº d'octet du fichier :	Ø	1	2	3	4	5	6	7	8	9	1Ø	11	12	13	14	15	16	17	18	19	2Ø	
Nº d'octet dans l'enregistrement :	Ø	1	2	3	4	Ø	1	2	3	4	Ø	1	2	3	4	Ø	1	2	3	4	Ø	
Rubrique :	Ø					Ø					Ø					Ø						
Enregistrement :			Ø					1					2					3				

Les termes « rubrique » et « enregistrement » seront explicités dans les chapitres 6, 7 et 8.

Les commandes OPEN, CLOSE, READ, WRITE et POSITION sont utilisées avec les deux types de fichiers, mais d'une manière quelque peu différente.

Les fichiers à accès séquentiel sont, d'une certaine manière, plus faciles à utiliser. Nous les verrons donc en premier lieu. L'utilisation des fichiers à accès direct sera vue au chapitre 8. L'annexe C contient des informations plus détaillées et plus techniques sur les différents types de fichiers.

Les fichiers à accès séquentiel : quelques exemples

Supposons que vous vouliez créer un fichier contenant une série de mots destinés à être utilisés dans un jeu. Voici deux paires de programmes qui traiteront un tel fichier. Le premier de chaque paire crée un fichier en mode texte, sur disquette. Le second de chaque paire permet une consultation des informations mémorisées sur disquette dans le fichier.

Ce programme crée un fichier nommé MOTS1, contenant les mots POMME, BANANE, CATALOG, DORMANT, AIGLE, FRUIT, OIE, CHAPEAU, GLAÇON.

```
 10  REM ECRITURE DU FICHIER MOTS1
 20  D$ = "": REM   CTRL-D
 30  PRINT D$; "OPEN MOTS1"
 40  PRINT D$; "WRITE MOTS1"
 50  PRINT "POMME"
 60  PRINT "BANANE"
 70  PRINT "CATALOG"
 80  PRINT "DORMANT"
 90  PRINT "AIGLE"
100  PRINT "FRUIT"
110  PRINT "OIE"
120  PRINT "CHAPEAU"
130  PRINT "GLAÇON"
140  PRINT D$; "CLOSE MOTS1"
150  END
```

A la ligne 30, la commande OPEN (précédée de l'habituel CTRL-D) *ouvre* un fichier en mode texte, nommé MOTS1, et l'inscrit dans le catalogue de la disquette, s'il n'y en avait pas précédemment un.

En ligne 40, la commande WRITE a pour effet de transformer tous les ordres PRINT qui la suivent : au lieu d'afficher les informations à l'écran, les ordres PRINT dirigent l'information vers le fichier en mode texte sur la disquette. Donc, dans ce programme, les lignes 50 à 130 enverront des mots sur la disquette, et non à l'écran.

La ligne 140 *ferme* le fichier, et arrête l'opération d'écriture.

Lors de l'exécution de ce programme, normalement, aucun message n'est affiché à l'écran. Mais si vous donnez une commande MON (voir chapitre 5), comme, par exemple,

```
MON C,I,O
```

(ou, simplement, MON C,O, puisqu'il n'y a pas de lecture de disquette dans ce programme), vous verrez l'affichage suivant à l'écran, lors de l'exécution :

```
OPEN MOTS1
WRITE MOTS1
POMME
BANANE
CATALOG
DORMANT
AIGLE
FRUIT
OIE
CHAPEAU
GLAÇON
CLOSE MOTS1
```

Vous venez de créer un fichier en mode texte, nommé MOTS1, sur votre disquette. Son nom sera précédé d'un T, lors de l'affichage du catalogue, pour indiquer son mode.

Le fichier est, en fait, une suite d'informations (en l'occurrence des mots), séparés par des retours chariot (RETURN). Un retour chariot est automatiquement généré après chaque PRINT non suivi d'une virgule ou

d'un point-virgule. Remarquez que le retour chariot est un *caractère* plutôt qu'une *action* (c'est en fait le caractère de valeur ASCII 13).

Chaque élément d'information suivi de son retour chariot est dénommé une *rubrique*. Une rubrique est mémorisée dans un fichier en mode texte par une série de valeurs ASCII. Le dernier caractère de chaque rubrique a la valeur 13 (retour chariot).

Le fichier MOTS1 est un fichier à accès séquentiel, parce que chaque rubrique est placée immédiatement après le caractère retour chariot de la rubrique précédente. Sur disquette, les rubriques sont de longueurs différentes : POMME prend six octets, un pour chaque lettre, plus un pour le RETURN; BANANE en prend sept, et ainsi de suite. Un fichier en mode texte, à accès séquentiel, est mémorisé sur la disquette sous la forme d'une longue suite ininterrompue de valeurs ASCII, une chaîne de rubriques, sans intervalles.

Naturellement, si vous avez créé un fichier sur disquette, il faut pouvoir le relire. Le programme suivant le permet. Il est écrit en BASIC étendu.

```
10  REM   CTRL-D
20  D$ = "" : REM CTRL-D
30  PRINT D$; "OPEN MOTS1"
40  PRINT D$; "READ MOTS1"
50  FOR I = 1 TO 9
60  INPUT A$(I)
70  NEXT I
80  PRINT D$; "CLOSE MOTS1"
90  END
```

En ligne 30, le fichier est ouvert. On ne peut pas, en effet, lire ou écrire un fichier sur disquette s'il n'a été préalablement ouvert.

En ligne 40, la commande READ transforme tous les INPUT ou les GET (BASIC étendu uniquement) qui suivent : au lieu que l'information soit acceptée à partir du clavier, elle le sera à partir de la disquette. Tout se passera comme si le lecteur tapait les réponses.

Une instruction INPUT fait accepter la totalité d'une rubrique, jusqu'au RETURN. Si une autre instruction INPUT suit, elle aura pour effet de faire accepter la prochaine rubrique, et ainsi de suite.

Les lignes 50 à 70, lorsqu'elles sont exécutées, ont pour effet de faire remplire la table A$ en A$(1), A$(2), A$(3),... des neuf rubriques lues à partir de la disquette. En ligne 80, le fichier est fermé.

Normalement, rien n'est visible à l'écran lors de l'exécution de ce programme. Mais si la commande MON C, I, O (ou simplement MON C,I) a été donnée préalablement, l'affichage suivant apparaîtra à l'écran :

```
OPEN MOTS1
READ MOTS1
?POMME
?BANANE
?CATALOG
?DORMANT
?AIGLE
?FRUIT
?OIE
?CHAPEAU
?GLAÇON
CLOSE MOTS1
```

Le point d'interrogation (?) est affiché avant chaque INPUT portant sur la disquette, exactement comme il le serait si l'INPUT portait sur le clavier.

Vous pouvez vérifier que tout a fonctionné comme prévu : tapez, par exemple,

```
PRINT A$(2), A$(9), A$(4)
```

et les mots BANANE, GLAÇON et DORMANT s'afficheront.

Si vous modifiez le programme MOTS1, pour enregistrer des mots différents, il vaut mieux, au préalable, effacer le fichier MOTS1 (DELETE MOTS1), faute de quoi vous pourriez vous retrouver avec un mélange des deux.

Dans l'exemple suivant, un fichier MOTS2 est créé, mais l'ensemble des neufs mots se trouve dans la même rubrique. Chaque mot est suivi d'une virgule, afin qu'un ordre INPUT avec de multiples variables (9, en l'occurrence) permettre d'accepter les mots un par un.

```
10  REM FICHIER MOTS2
20  D$ = "" : REM CTRL-D
30  PRINT D$; "OPEN MOTS2
40  PRINT D$; "WRITE MOTS2"
50  PRINT "POMME, BANANE, CATALOG,";
60  PRINT "DORMANT, AIGLE, FRUIT,";
70  PRINT "OIE, CHAPEAU, GLAÇON"
80  PRINT D$; "CLOSE MOTS2"
90  END
```

Remarquez que l'ordre PRINT en ligne 50 se termine par un point-virgule. Un point-virgule à la fin d'un ordre PRINT empêche l'émission automatique du retour chariot (RETURN). C'est pourquoi les caractères compris dans le prochain ordre PRINT feront partie de la même rubrique. L'ordre PRINT en ligne 60 se termine aussi par point-virgule. C'est seulement en ligne 70 qu'un ordre PRINT ne se termine pas par un point-virgule, permettant au retour chariot (RETURN) de s'effectuer normalement. La rubrique contient donc tous les caractères écrits en lignes 50, 60 et 70.

Habituellement, dans les ordres PRINT, les virgules commandent une tabulation automatique à l'écran. La fonction des virgules dans les ordres PRINT est modifiée lorsqu'il s'agit de l'écriture sur disquettes : dans ce cas, les virgules sont traitées comme des points-virgules, les caractères ne sont pas séparés des caractères du prochain PRINT, et aucun retour chariot automatique ne sera engendré.

Si vous exécutez le programme MOTS2, après MON C,I,O, vous verrez :

```
OPEN MOTS2
WRITE MOTS2
POMME, BANANE, CATALOG, DORMANT, AIGLE, FRUIT, OIE, CHAPEAU,
  GLAÇON
CLOSE MOTS2
```

Ce programme en BASIC étendu permet de lire le fichier MOTS2 :

```
10  REM LECTURE DU FICHIER MOTS2
20  D$ = "" : REM CTRL-D
30  PRINT D$; "OPEN MOTS2"
40  PRINT D$; "READ MOTS2"
50  INPUT A1$, A2$, A3$, A4$, A5$, A6$,
    A7$, A8$, A9$
80  PRINT D$; "CLOSE MOTS2"
90  END
```

Lors de son exécution (sous l'effet de la commande MON C,I,O), il affichera :

```
OPEN MOTS2
READ MOTS2
?POMME, BANANE, CATALOG, DORMANT, AIGLE, FRUIT, OIE, CHAPEAU,
  GLAÇON
CLOSE MOTS2
```

En BASIC simple, les virgules peuvent séparer des réponses multiples concernant des variables numériques, et non concernant des variables alphanumériques. C'est pourquoi, en BASIC simple, le programme « Lecture du fichier MOTS2 » affectera la totalité de la rubrique (neuf mots, huit virgules) à la variable A1$. Puis le message END OF DATA serait affiché lorsqu'il n'y aura plus de rubrique à assigner à A2$.

En BASIC étendu, on peut aussi utiliser l'instruction GET pour saisir des informations contenues dans un fichier en mode texte, caractère par caractère. Ceci permet de définir soi-même le caractère séparateur de rubriques.

L'exemple suivant est une autre façon de lire le fichier MOTS2, utilisant l'ordre GET.

```
10   CLEAR : REM GET DU FICHIER MOTS2
20   D$ = CHR$(4) : REM CTRL-D
30   R$ = CHR$(13) : REM RETURN
40   T$=CHR$(1): REM CTRL-A
50   PRINT D$; "OPEN MOTS2"
60   PRINT D$; "READ MOTS2"
70   I = I + 1
80   GET B$
90   IF B$ = "," THEN GOTO 70
100  IF B$ = R$ THEN GOTO 130
110  A$(I) = A$(I) +B$
115  PRINT T$; A$(I)
120  GOTO 80
130  PRINT R$; D$; "CLOSE MOTS"
140  END
```

En ligne 10, la commande CLEAR initialise toutes les variables (y compris I et tous les A$(I) à zéro.

En ligne 20, la constante CTRL-D est définie d'une manière qui n'est autorisée qu'en BASIC étendu. Cette méthode évite le caractère de contrôle invisible à l'écran.

En ligne 80, l'instruction GET permet de saisir un à un les caractères écrits sur disquette. Si le caractère saisi n'est ni une virgule, ni un retour chariot, la ligne 110 le place à la suite, dans la chaîne A$.

Lorsqu'une virgule est rencontrée, le premier mot est terminé. L'exécution est renvoyée en ligne 70, où l'incrémentation de I permet de commencer à composer un nouveau mot en A$(2). Et ainsi de suite, jusqu'à ce que le retour chariot soit rencontré, signalant la fin de la rubrique; enfin, le fichier est fermé en ligne 130.

Remarquez l'utilisation du CHR$(13) en ligne 30. On ne peut pas taper directement le caractère « retour chariot » dans un programme BASIC, car ce caractère indique la fin de ligne en BASIC. En BASIC étendu, le retour chariot est CHR$(13). — En BASIC simple, le retour chariot a une valeur ASCII de 141.

Lorsque les caractères sont saisis sur disquette par l'instruction GET, ils ne sont pas affichés à l'écran, même dans le mode MON C,I,O. La ligne 115 a été ajoutée pour vous permettre de voir les mots se constituer caractère par caractère.

Après une instruction BASIC étendu GET portant sur un fichier disquette en mode texte, trois problèmes se posent :

1) avec NOMON C,I,O, le premier caractère affiché (PRINT) après le GET, *n'apparaîtra pas* à l'écran;

2) avec MON C,I,O, le premier caractère affiché (PRINT) après le GET *apparaîtra* à l'écran;

3) que l'on soit en mode MON ou en mode NOMON, si une commande du DOS est le premier élément affiché (PRINT) après l'instruction GET, il est possible que la commande DOS ne s'exécute pas parce qu'elle n'a pas été précédée d'un retour chariot (RETURN).

Dans le programme GET DU FICHIER MOTS2, le caractère CTRL-A (T$) qui ne s'affiche pas est utilisé comme caractère non significatif, en ligne 115. Cette astuce permet de résoudre les deux premiers problèmes.

Pour résoudre le troisième problème, le caractère retour chariot (R$) a été affiché avant la commande DOS en ligne 130. C'est la même solution que celle adoptée pour résoudre le problème de l'utilisation de l'instruction TRACE, que nous avons vue précédemment.

Lorsque ce programme est exécuté, sous le mode MON C,I,O, l'affichage suivant apparaîtra sur une colonne, et non sur trois :

OPEN MOTS2	CATALO	O
READ MOTS2	CATALOG	OI
P	D	OIE
PO	DO	C
POM	DOR	CH
POMM	DORM	CHA
POMME	DORMA	CHAP
B	DORMANT	CHAPE
BA	A	CHAPEA
BAN	AI	CHAPEAU
BANA	AIG	G
BANAN	AIGL	GL
BANANE	AIGLE	GLA
C	F	GLAÇ
CA	FR	GLAÇO
CAT	FRU	GLAÇON
CATA	FRUI	
CATAL	FRUIT	

Enfin, voici un programme en BASIC étendu qui créée un fichier, MOTS3, comportant deux mots dans la première rubrique, trois mots dans la seconde, et quatre mots dans la troisième rubrique.

```
10  REM ECRITURE DU FICHIER MOTS3
20  D$ = CHR$ (4) : REM CTRL-D
30  PRINT D$; "OPEN MOTS3"
40  PRINT, D$; "WRITE MOTS3"
50  PRINT "POMME, BANANE"
60  PRINT "CATALOG, DORMANT, AIGLE"
70  PRINT "FRUIT, OIE, CHAPEAU, GLAÇON"
80  PRINT D$; "CLOSE MOTS3"
90  END
```

La première rubrique contient POMME, BANANE et a une longueur de 13 octets, 1 par caractère (la virgule compte aussi) plus le caractère retour chariot.
La seconde rubrique, CATALOG, DORMANT, AIGLE, a une longueur de 22 octets.
La troisième rubrique, FRUIT, OIE, CHAPEAU, GLAÇON, a une longueur de 25 octets.

L'affichage, lors de l'exécution (avec MON C,I,O) est le suivant :

```
OPEN MOTS3
WRITE MOTS3
POMME, BANANE
CATALOG, DORMANT, AIGLE
FRUIT, OIE, CHAPEAU, GLAÇON
CLOSE MOTS3
```

Voici un programme de lecture du programme MOTS3 :

```
10  REM LECTURE DU FICHIER MOTS3
20  D$ = CHR$(4) : REM CTRL-D
30  PRINT D$; "OPEN MOTS3"
40  PRINT D$; "READ MOTS3"
50  INPUT R$, S$
60  INPUT T$, U$, V$
70  INPUT W$, X$, Y$, Z$
80  PRINT D$; "CLOSE MOTS3"
90  END
```

Lors de l'exécution (avec MON C,I,O), vous verrez l'affichage suivant :

```
OPEN MOTS3
READ MOTS3
?POMME, BANANE
?CATALOG, DORMANT, AIGLE
?FRUIT, OIE, CHAPEAU, GLAÇON
CLOSE MOTS3
```

Les exemples de programmes donnés pour lire les fichiers MOTS1, MOTS2 et MOTS3 étaient soigneusement étudiés pour permettre la lecture du nombre exact de rubriques, et à l'intérieur de chaque rubrique, le nombre exact d'éléments. En général, un programme destiné à la lecture d'un fichier en mode texte doit être spécialement conçu pour ce fichier. Si le programme comporte une erreur, les résultats peuvent paraître aberrants. Par exemple, considérez le programme suivant, qui n'est pas correctement réalisé pour effectuer la lecture du fichier MOTS3 :

```
10  REM LECTURE DU FICHIER MOTS3 : BIS
20  D$ =: REM CTRL-D
30  PRINT D$; "OPEN MOTS3"
40  PRINT D$; "READ MOTS3"
50  INPUT R$, S$
60  INPUT T$, U$, V$
70  INPUT W$, X$, Y$
80  PRINT D$; "CLOSE MOTS3"
90  END
```

L'exécution (avec MON C,I,O) donnera l'affichage suivant :

```
OPEN MOTS3
READ MOTS3
?POMME, BANANE
?CATALOG, DORMANT, AIGLE
?FRUIT, OIE, CHAPEAU, GLAÇON
?EXTRA IGNORED
CLOSE MOTS3
```

L'instruction INPUT, en ligne 70, a eu pour effet de faire accepter la totalité de la rubrique contenant les mots FRUITS, OIE, CHAPEAU, GLAÇON. Les trois premiers furent acceptés dans les variables W$, X$ et Y$. Mais il n'y avait aucune variable pour recevoir le quatrième mot. C'est pourquoi le message EXTRA IGNORED est affiché.

Voici un autre programme qui n'est pas correctement réalisé pour la lecture du fichier MOTS3 :

```
10  REM LECTURE DU FICHIER MOTS3 TER
20  D$ = "" : REM CTRL-D
30  PRINT D$; "OPEN MOTS3"
40  PRINT D$; "READ MOTS3"
50  INPUT R$, S$
60  INPUT T$, U$, V$, W$
70  INPUT X$, Y$, Z$
80  PRINT D$; "CLOSE MOTS3"
90  END
```

Et voici l'affichage lors de l'exécution, avec MON C,I,O :

```
OPEN MOTS3
READ MOTS3
?POMME, BANANE
?CATALOG, DORMANT, AIGLE
??FRUIT, OIE, CHAPEAU, GLAÇON
?EXTRA IGNORED
?
END OF DATA
BREAK IN 70
```

En ligne 60, la rubrique contenant CATALOG, DORMANT, AIGLE est lue. Les trois mots sont acceptés dans les variables T$, U$ et V$. Une quatrième variable était prévue en ligne 60. Aussi est-ce la totalité de la rubrique suivante que le DOS a tenté d'y mettre. Etant donné qu'il n'y avait place que pour un seul mot, c'est le premier mot de la prochaine rubrique FRUIT, OIE, CHAPEAU, GLAÇON qui est mis dans W$, soit le mot FRUIT. Il n'y a pas d'autre variable pour recevoir les mots suivants; c'est la raison pour laquelle le message EXTRA IGNORED est affiché.

Lorsque l'exécution arrive en ligne 70, il n'y a plus de rubrique à lire. C'est pourquoi le message END OF DATA est affiché.

Nous verrons plus loin un exemple plus général de création et de lecture d'un fichier en mode text : ECRITURE TEXTE et LECTURE TEXTE.

L'ouverture et la fermeture des fichiers à accès séquentiel

Les fichiers séquentiels sont à utiliser lorsque l'information doit être consultée de manière linéaire, partant du début et allant jusqu'à la fin, et lorsque le nombre de mouvements (modifications, mises à jour, suppressions) est peu élevé.

Pour créer un fichier séquentiel, les commandes DOS :

```
OPEN
WRITE
PRINT
CLOSE
```

sont utilisées dans l'ordre ci-dessus, mais pas nécessairement l'une après l'autre.

Pour lire un fichier séquentiel, les commandes DOS :

```
OPEN
READ
INPUT
CLOSE
```

sont utilisées dans l'ordre ci-dessus, mais pas nécessairement l'une après l'autre.

Avant d'utiliser un fichier, que ce soit pour la lecture ou pour l'écriture, il est nécessaire au préalable de l'ouvrir (OPEN). Lorsque les opérations sont finies avec un fichier, il faut le fermer (CLOSE).

Un fichier qui a été ouvert doit être clôturé !

Si l'on omet la clôture d'un fichier, alors que des opérations d'écriture ont été effectuées, il est possible que le fichier ne soit plus exploitable, et donc que les informations soient perdues.

La syntaxe pour ces commandes est :

```
OPEN f   [,Ss]   [,Dd]   [,Vv]
CLOSE    [f]
```

Exemples :

```
OPEN SESAME
OPEN COMPTA,D2,S7
CLOSE PUB
```

La commande OPEN alloue une mémoire tampon pour le fichier f (595 octets), et initialise les pointeurs pour commencer la lecture (ou l'écriture) au début du fichier. La commande OPEN initialise aussi les numéros de port et de lecteur devant être utilisés pour les commandes READ et WRITE à venir, afférentes à ce fichier.

La commande CLOSE désalloue la mémoire tampon, et rend donc les 595 octets à la mémoire utilisateur. Si aucun fichier n'est spécifié dans la commande CLOSE, tous les fichiers ouverts sont fermés, sauf ceux qui sont utilisés par une commande EXEC (voir chapitre 7, pour la commande EXEC).

Lorsque la commande OPEN s'exécute, elle contrôle que le fichier concerné, s'il existe déjà, est fermé, avant de procéder à l'ouverture. S'il ne l'est pas, la commande OPEN ferme d'abord le fichier.

Remarquez que la commande CLOSE ne comporte pas de numéro de port, ou de numéro de lecteur. Si une commande

```
CLOSE STOCK
```

est donnée, tout fichier nommé STOCK sera fermé, quel que soit son numéro de port ou de lecteur.

Si une commande

```
CLOSE
```

est donnée, tous les fichiers, sur tous les numéros de ports, et sur tous les lecteurs seront fermés (sauf ceux utilisés par une commande EXEC).

Dans certaines circonstances, il est possible que vous souhaitiez effacer un fichier f, sans avoir la certitude qu'il existe sur la disquette. Ce cas peut se présenter particulièrment si vous désirez créer un tout nouveau fichier portant le même nom qu'un fichier déjà existant éventuellement antérieurement. En effet, si le fichier précédent n'est pas effacé, et si le nouveau fichier créé est plus court que le précédent, une partie de l'ancien fichier se retrouvera à la fin du nouveau.

Supposons qu'un programme utilise un fichier de travail, et chaque fois que l'on entame une nouvelle opération spécifique dans ce programme, on désire repartir avec un fichier de travail tout neuf. Si l'on se contente d'écrire dans le programme :

```
DELETE TRAVAIL
```

tout ira très bien si le fichier TRAVAIL existe. Mais s'il n'existe pas encore, le message

```
FILE NOT FOUND
```

sera affiché et l'exécution sera arrêtée.

Pour éviter ce problème, il suffit de donner deux commandes OPEN : l'une avant la commande DELETE, l'autre après.

Par exemple :

```
10  D$ = "" : REM CTRL-D
20  PRINT D$; "OPEN TRAVAIL"
30  PRINT D$; "DELETE TRAVAIL"
40  PRINT D$; "OPEN TRAVAIL"
50  REM SUITE DU PROGRAMME
```

L'écriture des fichiers séquentiels

Voici un autre exemple en BASIC étendu de création d'un fichier nommé ECHANTILLON et qui contient trois chaînes de caractères et dix numéros :

```
  5  REM ECRITURE DE ECHANTILLON
 10  D$ = CHR$(4) : REM CTRL-D
 20  PRINT D$; "OPEN ECHANTILLON"
 30  PRINT D$; "DELETE ECHANTILLON"
 40  PRINT D$; "OPEN ECHANTILLON"
 50  PRINT D$; "WRITE ECHANTILLON"
 60  PRINT "PREMIER TEXTE" : PRINT "SECOND TEXTE"
 70  PRINT "TROISIEME TEXTE"
 80  FOR J = 1 TO 10
 90  PRINT J
100  NEXT J
110  PRINT D$; "CLOSE ECHANTILLON"
120  END
```

Avec MON C,I,O, l'affichage apparaîtra lors de l'exécution :

```
OPEN ECHANTILLON
DELETE ECHANTILLON
OPEN ECHANTILLON
WRITE ECHANTILLON
PREMIER TEXTE
SECOND TEXTE
TROISIEME TEXTE
1
2
3
4
5
6
7
8
9
10
CLOSE ECHANTILLON
```

Une fois qu'une commande WRITE a été donnée, tous les ordres PRINT qui suivent dirigent les caractères vers la disquette, et non vers l'écran.
Cette situation prend fin lorsque le prochain

```
PRINT D$
```

(si D$ contient un CTRL-D) est rencontré, qu'il soit suivi ou non d'une commande DOS.

ATTENTION!
Une commande *WRITE est en cours* si elle a été lancée et que l'ordre PRINT D$ n'a pas encore été rencontré.

Si une commande WRITE est en cours, et qu'une instruction INPUT est donnée, par exemple : INPUT X$, le point d'interrogation normalement engendré par l'ordre INPUT à l'écran, sera écrit sur disquette, dans le fichier. L'ordre aura ensuite le même effet que le PRINT D$, c'est-à-dire qu'il terminera la commande WRITE.

Si l'instruction INPUT était de la forme :

```
INPUT "QUEL EST VOTRE NOM?"; X$
```

c'est la totalité du message, suivi du point d'interrogation, qui serait écrit sur disquette, avant que la commande WRITE ne soit terminée.

Le même danger existe pour les *messages d'erreur du BASIC*. Si, pour une raison quelconque, un message d'erreur du BASIC est émis alors qu'une commande WRITE est en cours, le message d'erreur sera écrit dans la dernière rubrique du fichier, sur disquette, avant que la commande WRITE ne soit terminée.

La syntaxe pour la commande WRITE, pour une utilisation en accès séquentiel est :

```
WRITE f
```

Exemples :

```
WRITE ARTICLES
WRITE COMMANDES
```

Le programme ECHANTILLON développé au début de ce chapitre est une illustration simple des connaissances de base nécessaires pour créer un fichier séquentiel en mode texte. Un programme quelque peu plus général, dont le nom est MAKE TEXT (faire du texte) est sur la disquette Maître fournie avec le lecteur.

MAKE TEXT est un programme qui vous permet de créer un fichier séquentiel en mode texte, contenant jusqu'à 100 chaînes de caractères. Chaque chaîne peut comporter jusqu'à 239 caractères. Essayez-le :

```
LOAD MAKE TEXT
```

Une liste du programme vous donnera :

```
  5  REM MAKE TEXT
 10  DIM A$(100) : I = 0
 20  D$ = CHR$(4) : REM CTRL-D
 30  HOME : TEXT
 40  PRINT "YOU GET TO TYPE ONE STRING AT TIME"
 50  PRINT "A STRING MAY HAVE UP TO 239 CHARACTERS"
 60  PRINT THIS PROGRAMM LETS YOU WRITE TEXT FILES"
 70  PRINT : I = I + 1
 80  PRINT "(PRESS RETURN KEY TO QUIT)"
 90  PRINT "TYPE STRING #"; I; " : ";
100  INPUT ""; A$(I)
110  IF A$(I) <> "" GOTO 60
120  PRINT
130  INPUT "WHAT FILE NAME?"; N$
140  PRINT D$; "OPEN"; N$
150  PRINT D$; "DELETE"; N$
160  PRINT D$; "OPEN"; N$
170  PRINT D$; "WRITE"; N$
180  PRINT I - 1
190  FOR J = 1 TO I - 1
200  PRINT A$(J)
210  NEXT J
220  PRINT D$; "CLOSE"; N$
```

Une fois que le programme est chargé en mémoire, sauvegardez-le sur une disquette qui n'est pas protégée contre l'écriture. Puis tapez :

```
MON C,I,O
```

afin de visualiser les opérations portant sur la disquette à l'écran. Tapez RUN. L'affichage suivant apparaîtra :

```
YOU GET TO TYPE ONE STRING AT A TIME
A STRING MAY HAVE UP TO 239 CHARACTERS
THIS PROGRAMM LETS YOU WRITE TEXT FILES
(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING # 1 :
```

ce qui signifie :

Vous taperez une suite de caractères à la fois.
Une suite peut comporter jusqu'à 239 caractères.
Ce programme vous permet d'écrire des fichiers en mode texte.
(Pour interrompre, frapper d'abord la touche 'RETURN').
Tapez la suite # 1 :

Tapez des chaînes de caractères (100 au maximum). Soyez attentif à ne pas utiliser de virgule ou de point-virgule dans vos suites de caractères, étant donné que la saisie est réalisée dans le programme par l'instruction INPUT. A la fin, il suffit de frapper la touche RETURN. Le programme vous demandera alors le nom du fichier que vous avez choisi, par la question :

```
WHAT FILE NAME ?
```

Puis l'affichage suivant apparaîtra à l'écran :

```
OPEN f
DELETE f
OPEN f
WRITE f
```

où f est le nom de fichier fourni. Le nombre de chaîne de caractères que vous avez entrées précédemment dans le fichier s'affichera ensuite. Puis vous verrez les suites de caractères, et enfin la commande :

```
CLOSE f
```

Voici un exemple d'exécution (avec MON C,I,O) du programme MAKE TEXT :

```
THIS PROGRAM LETS YOU WRITE TEXT FILES
YOU GET TO TYPE ONE STRING AT A TIME
A STRING MAY HAVE UP TO 239 CHARACTERS
(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING # 1: HERE'S STRING 1
(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING # 2: AND MY SECOND STRING
(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING # 3: ONE WE GO
(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING # 4: ENOUGH ALREADY!
```

```
(TO QUIT, PRESS RETURN KEY FIRST)
TYPE STRING # 5:
WHAT FILE NAME? TEST
OPEN TEST
DELETE TEST
OPEN TEST
WRITE TEST
4
HERE'S STRING 1
AND MY SECOND STRING
ONE WE GO
ENOUGH ALREADY!
CLOSE TEST
```

Si vous ouvrez (OPEN) un fichier en mode texte qui existe déjà et puis appliquez la commande WRITE (sans d'abord effacer (DELETE) le fichier et l'ouvrir (OPEN) à nouveau), au moins une partie du fichier sera surimprimée. Le résultat sera un mélange de données imprimées (PRINT) aux deux occasions, à moins que tous les caractères de l'ancien fichier ne soient surimprimés. Les nouveaux caractères que vous avez imprimés au fichier cette fois apparaîtront d'abord et puis la partie de l'ancien fichier qui n'a pas été surimprimée. Pour supprimer tous les caractères de l'ancien fichier, ouvrez (OPEN) et effacez (DELETE) l'ancien fichier avant de l'ouvrir (OPEN) à nouveau. (Dans le programme MAKE TEXT (faire du texte), les lignes 140 et 150 s'occupent du "nettoyage" de n'importe quel fichier en mode texte ancien du même nom). Pour éviter qu'un programme ne surimprime un fichier, fermez (LOCK) le fichier.

La lecture des fichiers à accès séquentiel

La commande DOS READ permet de lire un fichier en mode texte. Dès qu'une commande READ est donnée, toutes les instructions INPUT (ou GET, en BASIC étendu uniquement), portent sur la disquette, et non sur le clavier.

Ce programme, écrit en BASIC étendu, permet de lire le fichier ECHANTILLON, créé dans un exemple précédent. La commande READ, comme la commande WRITE, doit être précédée d'une commande OPEN. Lorsque toutes les lectures sont terminées, le fichier doit aussi être fermé, par une commande CLOSE.

```
 5  REM LECTURE DU FICHIER ECHANTILLON
10  D$ = CHR$(4) : REM CHR$(4) EST LE CTRL-D
20  PRINT D$; "OPEN ECHANTILLON"
30  PRINT D$; "READ ECHANTILLON"
40  INPUT A$,B$,C$
50  FOR I = 1 TO 10
60  INPUT W
70  NEXT I
80  PRINT D$; "CLOSE ECHANTILLON"
```

La commande OPEN doit précéder la première commande READ d'un programme, et l'instruction INPUT (ou GET) doit la suivre. Les commandes OPEN et READ doivent porter sur le même fichier.

Lors de l'exécution du programme précédent (avec MON C,I,O), vous verrez l'affichage suivant :

```
OPEN ECHANTILLON
READ ECHANTILLON
?PREMIER TEXTE
??SECOND TEXTE
??TROISIEME TEXTE
?1
?2
?3
?4
?5
?6
?7
?8
?9
?10
CLOSE ECHANTILLON
```

Le programme a été spécialement écrit pour le fichier ECHANTILLON : il est adapté à un fichier comportant trois suites de caractères (A$, B$ et C$ en ligne 40) et dix nombres (W en ligne 60). Deux points d'interrogation sont affichés lorsque B$ et C$ sont acceptés, parce que des retours chariot séparaient les suites, alors qu'elles sont acceptées dans la même instruction INPUT.

Une commande READ est terminée par une quelconque autre commande du DOS dans un ordre PRINT. Une commande DOS comportant simplement :

```
PRINT D$
```

peut être utilisée pour signifier la fin d'une commande READ, et le retour au clavier pour toutes les instructions INPUT qui la suivent.

Une commande PR # ou IN # a aussi pour effet de terminer une commande READ.

La syntaxe pour la commande READ est :

```
READ f
```

Exemples :

```
READ COMMANDES
READ ARTICLES
```

Si une interruption causée par un CTRL-C intervient alors qu'une commande READ est en cours, dans un programme écrit en BASIC étendu, elle engendrera une suite de REENTER. Pour l'éviter, utiliser la touche RESET pour interrompre le programme.

Un programme, écrit en APPLESOFT, et qui permet de lire le fichier MAKE TEXT, est enregistré sur la disquette Maître du système. Pour l'exécuter, insérez la disquette Maître dans le lecteur, et tapez :

```
LOAD RETRIEVE TEXT
```

puis sauvegardez-le sur la disquette sur laquelle MAKE TEXT est déjà enregistré. Une liste du programme vous donnera :

```
 10  D$ = CHR$ (4) : REM CTRL-D
 12  PRINT "THIS PROGRAM RETRIEVES TEXT FILES"
 14  PRINT "CREATED BY 'MA TEXT' PROGRAM"
 16  PRINT "MON C,I,O IS IN EFFECT"

 18  PRINT
 20  INPUT "NAME OF TEXT FILE ?"; Z$
 22  PRINT D$; "MON C,I,O"
 24  PRINT
 30  PRINT D$; "OPEN"; Z$
 40  PRINT D$; "READ"; Z$
 50  INPUT I
 55  DIM A$(I)
 60  FOR J = 1 TO I
 70  :INPUT A$(J)
 80  NEXT J
 90  PRINT D$; "CLOSE"; Z$
100  PRINT D$; "NOMON C,I,O"
```

Lors de l'exécution avec le fichier précédemment créé, le texte suivant sera affiché :

```
THIS PROGRAM RETRIEVES TEXT FILES
CREATED BY 'MAKE TEXT' PROGRAM
MON C,I,O IS IN EFFECT

NAME OF TEXT FILE ? TEST

OPEN TEST
READ TEST
?4
?VOICI LE PREMIER TEXTE
?VOICI LE SECOND MESSAGE
?ENCORE UN!
CLOSE TEST
NOMON, C,I,O
```

Append et position

Les commandes APPEND et POSITION permettent d'ajouter des informations à la fin d'un fichier séquentiel en mode texte, et d'accéder à une information placée à un endroit déterminé du fichier.

APPEND permet de poursuivre l'écriture d'un fichier séquentiel, à la suite de la dernière fiche enregistrée. La commande OPEN positionne le pointeur du DOS à l'octet 0 du fichier. La commande APPEND réalise l'ouverture (OPEN) du fichier, et positionne le pointeur à la prochaine place disponible à la suite des enregistrements déjà écrits.

Le prochain exemple crée un fichier nommé TESTER qui contient deux suites de caractères : TEST 0 et TEST 1 :

```
 5  REM MAKE TESTER
10  D$ = CHR$(4) : REM CTRL-D
20  PRINT D$; "OPEN TESTER"
30  PRINT D$; "DELETE TESTER"
```

```
40  PRINT D$; "OPEN TESTER"
50  PRINT D$; "WRITE TESTER"
60  PRINT "TEST 0"
70  PRINT "TEST 1"
80  PRINT D$; "CLOSE TESTER"
```

Le programme suivant permet d'ajouter trois enregistrements au fichier TESTER, qui en comporte déjà 2 :

```
 5  REM APPEND TESTER
10  D$ = CHR$(4) : REM CTRL-D
20  PRINT D$; "APPEND TESTER"
30  PRINT D$; "WRITE TESTER"
40  PRINT "TEST 2"
50  PRINT "TEST 3"
60  PRINT "TEST 4"
70  PRINT D$; "CLOSE TESTER"
```

Le programme suivant lit le fichier TESTER :

```
10  REM RETRIEVE TESTER
20  D$ = CHR$(4) : REM CTRL-D
30  PRINT D$; "OPEN TESTER"
40  PRINT D$; "READ TESTER"
50  FOR I = 1 TO 5
60  INPUT A$
70  NEXT I
80  PRINT D$; "CLOSE TESTER"
```

La commande APPEND doit être suivie d'une commande WRITE (une tentative de lecture donnerait le message d'erreur END OF DATA).

La syntaxe pour la commande APPEND est :

```
APPEND f (,Ss) (,Dd) (,Vv)
```

Bien que APPEND soit utilisé exclusivement lorsque l'écriture sur disquette est prévue, le DOS n'engendre pas le message

```
FILE LOCKED
```

si le fichier en question est verouillé. Ce message ne sera affiché que lors de la tentative d'écriture.

La commande POSITION permet de lire, ou d'écrire, un enregistrement dans un fichier séquentiel en mode texte, à un endroit repréré par un numéro de rubrique.

La syntaxe pour la commande POSITION est :

```
POSITION f (, RP)
```

où RP est la position *relative au pointeur.*

Cette commande fait avancer (jamais reculer) le pointeur de positionnement du DOS à l'intérieur du fichier, de P rubriques. Si P = 0, la prochaine commande READ ou WRITE concernera la rubrique normalement à venir.
Si P = 1, la commande READ ou WRITE «saute» une rubrique, et traite celle qui suit. Si P = 2, la commande READ ou WRITE «saute» deux rubriques, et traite celle qui suit. Et ainsi de suite. Si le fichier ne

comporte pas de rubrique correspondante à la position relative spécifiée, le message d'erreur END OF DATA sera affiché, et l'exécution sera interrompue.

POSITION doit porter sur un fichier préalablement ouvert. La commande OPEN positionne le pointeur de fichier à la rubrique 0. Si POSITION est utilisé immédiatement après OPEN, et seulement dans ce cas, la position relative correspond à la position réelle, ou absolue, du numéro de rubrique.

Comme toute autre commande DOS, POSITION termine une commande READ ou WRITE. La commande POSITION doit donc être donnée avant la commande READ ou WRITE associée.

Lors de l'exécution de la commande POSITION, le fichier est exploré, à la recherche du RPième RETURN. Si, au cours de cette exploration, un octet «vide» (00) est rencontré, le message END OF DATA sera immédiatement affiché (sans nécessiter une instruction INPUT ou GET). Il est donc important qu'un fichier séquentiel ne comporte pas de fiche «vide».

Voici un programme qui utilise la commande POSITION pour lire différentes rubriques du fichier TESTER, créé antérieurement par les programmes MAKE TESTER et APPEND TESTER :

```
 10  REM POSITION TESTER
 20  D$ = CHR$(4) : REM CTRL-D
 30  PRINT D$; "OPEN TESTER"
 40  PRINT D$; "POSITION TESTER,R2"
 50  PRINT D$; "READ TESTER"
 60  INPUT A$
 70  PRINT D$; "POSITION TESTER,R1"
 80  PRINT D$; "READ TESTER"
 90  INPUT B$
100  PRINT D$; "OPEN TESTER"
110  PRINT D$; "POSITION TESTER,R3"
120  PRINT D$; "READ TESTER"
130  INPUT C$
140  INPUT E$
150  PRINT D$; "CLOSE TESTER"
```

Lors de l'exécuton (sous l'effet de MON C,I,O), vous verrez l'affichage suivant :

```
OPEN TESTER
POSITION TESTER,R2
READ TESTER
?TEST2
POSITION TESTER,R1
READ TESTER
?TEST4
OPEN TESTER
POSITION TESTER,R3
READ TESTER
?TEST3
?TEST4
CLOSE TESTER
```

Si le résulat vous surprend, rappelez-vous qu'il s'agit du numéro de rubrique relatif par rapport à la position du pointeur, que celui-ci est initialisé à zéro par la commande OPEN, et que chaque instruction INPUT portant sur le fichier fait progresser le pointeur de 1.

La lecture et l'écriture partielles

Les informations qui suivent ne sont pas destinées aux débutants : les fichiers séquentiels peuvent parfaitement être utilisés alors même que l'on ignore l'utilisation des paramètres, développée ci-après.

Les commandes DOS READ et WRITE peuvent être utilisées avec un paramètre donnant le numéro d'octet (byte) à partir duquel on désire faire commencer la commande. Le plus difficile est de savoir exactement à

partir de quel octet on désire commencer. Cela implique que l'on sache exactement quelle information y a été enregistrée, y compris tous les espaces, virgules, retours chariot et autres caractères (chaque octet contient une valeur ASCII correspondante à un caractère). Le problème est encore plus difficile avec la commande WRITE qu'avec la commande READ, car il faut savoir où s'arrêter, c'est-à-dire le nombre exact de caractères à écrire.

Le paramètre B (numéro d'octet) est une valeur absolue à partir de l'octet Ø du début du fichier, si le paramètre R (numéro de rubrique) n'est pas spécifié.

Si, par contre, le paramètre R est spécifié, le paramètre B est une valeur relative à partir de l'octet Ø de la Rième rubrique.

La commande

```
WRITE PROSPECTS,B27
```

positionne le pointeur du fichier au 28e octet du fichier PROSPECTS (le premier octet est référencé Ø). Les caractères dirigés vers la disquette par une instruction PRINT suivant cette commande WRITE, remplaceront les caractères préalablement écrits dans le fichier, à partir du 28e, et jusqu'à épuisement du nombre de caractères écrits par l'ordre PRINT.

Attention : l'écriture n'est pas limitée par la longueur de la rubrique préalablement écrite dans le fichier. Si le nombre de caractères écrits ne va pas jusqu'à la fin de la rubrique, il y aura en fait deux rubriques à cet endroit : les caractères qui viennent d'être écrits d'une part, et la queue de l'ancienne rubrique, d'autre part.

Si, par contre, les caractères écrits vont plus loin que la fin de la rubrique, c'est le début de la rubrique suivante qui s'en trouve «écrasé». Nous aurons donc une rubrique plus longue que précédemment, suivie d'une rubrique plus courte.

Il est aussi possible d'écrire en commençant à l'octet spécifié par B qui est immédiatement après le dernier octet écrit dans le fichier séquentiel. Bien entendu, une tentative de lecture de cet octet donnerait le message END OF DATA et l'exécution du programme serait arrêtée.

La syntaxe pour cette commande est :

```
WRITE f (,Bb)
```

où B spécifie le numéro d'octet à l'intérieur du fichier, à partir duquel la prochaine instruction PRINT placera les caractères dans le fichier. La valeur par défaut de B est Ø, c'est-à-dire le numéro du premier octet du fichier. Le paramètre B est une valeur réelle (et non relative) de numéro d'octet.

Le paramètre B peut concerner un octet placé en amont, ou en aval, de la position du pointeur de fichier, à un moment donné.

De même, la commande :

```
READ PROSPECTS,B32
```

positionne le pointeur de fichier au 33e octet (le premier octet est référencé 0) du fichier PROSPECTS. Une instruction INPUT intervenant ensuite aura pour effet de faire accepter tous les caractères, à partir du 33e, et jusqu'à ce qu'un retour chariot soit rencontré. Si le 33e octet ne concerne pas le premier caractère d'une rubrique, la rubrique lue sera incomplète.

La syntaxe pour cette commande est :

```
READ f (,Bb)
```

où le paramètre B spécifie le numéro à l'intérieur du fichier à partir duquel la lecture débutera. La valeur par défaut de B est 0 (le numéro du premier octet dans le fichier). Le paramètre B peut concerner un octet placé en amont ou en aval de la position du pointeur de fichier, à un moment donné.

Le programme suivant positionne le pointeur de fichier à l'octet 14 (le 15e octet du fichier, dans le programme MAKE TESTER. Puis il y écrit la suite de caractères "APPLE II".
Remarquez la séquence habituelle des commandes du DOS : OPEN, WRITE, PRINT, CLOSE :

```
 5  REM ECRITURE A PARTIR D'UN OCTET DETERMINE
10  D$ = CHR$(4) : REM CTRL-D
20  D$; "OPEN TESTER"
30  PRINT D$; "WRITE TESTER,B14"
40  PRINT "APPLE II"
50  PRINT D$; "CLOSE TESTER"
```

Si vous exécutez le programme RETRIEVE TESTER, pour lire le fichier TESTER, vous constaterez les modifications apportées à ce fichier par le programme précédent :

La rubrique contenant APPLE II a complétement recouvert la rubrique qui contenait précédemment TEST 2, ainsi que le premier caractère de la rubrique suivante TEST 3. Le fichier ne comporte donc plus que quatre rubriques, au lieu de cinq. La troisième est dorénavant :

```
APPLE TEST 3
```

Comme cinq instructions INPUT sont prévues dans le programme RETRIEVE TESTER, lorsque la dernière s'exécute, alors qu'il n'y a plus de rubrique à lire dans le fichier, le message END OF DATA s'affiche, et l'exécution est ininterrompue.

Le programme suivant positionne le pointeur de fichier à l'octet 18 dans le fichier TESTER, qui vient d'être modifié par le programme précédent. Puis ce programme permet la lecture des caractères, à partir du pointeur jusqu'au prochain retour chariot.

```
 5  REM LECTURE A PARTIR D'UN OCTET DETERMINE
10  D$ = CHR$(4) : REM CTRL-D
20  PRINT D$; "OPEN TESTER"
30  PRINT D$; "READ TESTER,B18"
40  INPUT A$
50  PRINT D$; "CLOSE TESTER"
```

Essayez de deviner le résultat, avant d'exécuter ce programme.

CHAPITRE 7
Les procédures catalogues

70 Le contrôle du système par un fichier en mode texte : EXEC
71 La création d'une procédure cataloguée
72 La transformation d'un programme en un fichier en mode texte
73 La conversion de routines langage machine en BASIC
74 La commande MAXFILES et le BASIC simple
74 La commande EXEC

Le contrôle du système par un fichier en mode texte : EXEC

La commande EXEC est semblable à la commande RUN, sauf qu'un fichier sur disquette est utilisé par la commande EXEC. Ce fichier contient des commandes DOS et des instructions BASIC, et se substitue à un opérateur qui donnerait des ordres successifs au micro-ordinateur à partir du clavier. Nous nommerons ce système une procédure cataloguée.

Un programme de démonstration de la commande EXEC est enregistré sur la disquette Maître du système. Tapez

LOAD EXEC DEMO

puis sauvegardez ce programme sur une disquette qui n'est pas protégée contre l'écriture.

Exécutez ce programme. L'affichage suivant apparaîtra sur l'écran :

« EXEC DEMO »
THIS PROGRAM CREATES A SEQUENTIAL TEXT FILE NAMED "DO'ER"
CONTAINING SEVERAL STRINGS, EACH A LEGAL APPEL II COMMAND.
WHEN YOU TYPE
EXEC DO'ER
THEN THE COMMANDS IN FILE DO'ER TAKE CONTROL OF YOUR COMPUTER.
EACH COMMAND WILL BE EXECUTED JUST AS IF IT HAD BEEN TYPED AT
THE KEYBOARD. THE DOS MANUAL DESCRIBES THIS PROGRAM IN MORE
DETAIL.
« HAPPY EXECUTING »
PRESS THE SPACE BAR TO MAKE THIS PROGRAM CREATE THE FILE DO'ER.
IF YOU WISH TO STOP THIS PROGRAM NOW, YOU MAY PRESS THE ESC KEY.

Ce message signifie :
Ce programme crée un fichier séquentiel en mode texte, nommé DO'ER, contenant plusieurs suites de caractères, chaque suite étant une commande APPLE II autorisée.
Lorsque vous tapez

DO'ER

les commandes dans le fichier DO'ER prennent le contrôle de votre micro-ordinateur. Chaque commande sera exécutée correctement comme si elle avait été tapée au clavier. Le DOS Manuel décrit ce programme plus en détail.
Bonne exécution! Appuyez sur la barre d'espacement pour que ce programme crée le fichier DO'ER. Si vous désirez interrompre maintenant, vous pouvez le faire en frappant la touche ESC.

Si vous frappez la barre d'espacement, vous verrez le fichier DO'ER s'écrire sur disquette. Si vous tapez :

EXEC DO'ER

votre micro-ordinateur exécutera les commandes contenues dans le fichier DO'ER.

Voici un résumé des principales opérations que comporte la procédure cataloguée DO'ER :
– une commande MON C,I,O est donnée, pour que tout soit visible;
– un programme comportant trois lignes seulement est décrit, et sauvegardé sur disquette sous le nom NEW PROGRAM!!;
– ce programme est ensuite « listé »;

– une boucle FOR-NEXT est utilisée comme boucle temps, afin que la liste du programme puisse être vue et lue;
– la commande INT permet de passer en BASIC simple;
– le programme COLOR DEMOS est chargé, et listé;
– le sous-programme -155 est appelé pour passer en mode Moniteur. Quelques instructions en langage machine sont exécutées avant que la commande FP n'appelle le BASIC étendu;
– à partir du BASIC étendu, une commande MON C;I,O est donnée, puis le programme NEW PROGRAM!! est exécuté, puis modifié, puis listé (une autre boucle temps permet de voir la liste), puis sauvegardé sous le nouveau nom EVEN MORE RECENT PROGRAM!!;
– enfin le programme NEW PROGRAM!! est effacé (DELETE) et le CATALOG, comprenant le dernier programme sauvegardé est affiché.

Tout ceci s'est effectué sans que vous ayez à frapper une seule touche du clavier.

La création d'une procédure cataloguée

Voici un exemple qui illustre, pas à pas, la réalisation d'une procédure cataloguée par la création d'un fichier EXEC nommé FAITES-LE, contenant les commandes suivantes :

```
LIST 20, 50
RUN N'IMPORTE QUOI
CATALOG
```

Tout d'abord, il convient de créer, et de sauvegarder, un programme en BASIC étendu nommé N'IMPORTE QUOI, afin de l'utiliser dans l'exemple de procédure cataloguée :

```
 5  REM N'IMPORTE QUOI
10  PRINT "CE PROGRAMME NE FAIT RIEN"
```

Le programme suivant, nommé CRÉATION PROCÉDURE, va créer le fichier FAITES-LE, comportant les commandes destinées à être ultérieurement exécutées par la procédure cataloguée. *Remarquez que les commandes qui sont écrites dans le fichier FAITES-LE ne sont pas précédées du CTRL-D.*

```
 5  REM CRÉATION PROCÉDURE
10  D$ = CHR$ (4) : REM CTRL-D
20  PRINT D$; "OPEN FAITES-LE"
30  PRINT D$; "WRITE FAITES-LE"
40  PRINT "LIST 20, 50"
50  PRINT "RUN N'IMPORTE QUOI"
60  PRINT "CATALOG"
70  PRINT D$; "CLOSE FAITES-LE"
```

Après la sauvegarde du programme nommé CRÉATION PROCÉDURE, tapez :

```
RUN CRÉATION PROCÉDURE
```

pour créer le fichier nommé FAITES-LE.

Tapez la commande :

```
EXEC FAITES-LE
```

pour exécuter les différentes commandes de la procédure, l'une après l'autre, exactement comme si elles avaient été tapées au clavier.

La transformation d'un programme en un fichier en mode texte

Voici une autre utilisation très utile de la commande EXEC qui vous permet de transformer provisoirement un programme en un fichier en mode texte. Grâce à cette transformation, il sera notamment possible de :

* traduire un programme écrit en BASIC simple en BASIC étendu;
* renuméroter une partie de programme, et l'insérer dans un autre programme;
* insérer des routines dans un programme, à partir d'un fichier comportant tous vos sous-programmes;
* joindre un programme à la suite d'un autre;
* « réparer » des programmes qui sont devenus partiellement illisibles.

Le programme suivant permettra la transformation d'un programme en un fichier. Les numéros de ligne 2270 et 5130 qui suivent l'instruction LIST en ligne 6 du programme CAPTURE doivent être remplacés par les numéros de ligne du programme que l'on désire transformer. Le nom du fichier séquentiel créé contenant la liste du programme est LISTING.

```
1  REM CAPTURE
2  D$ = CHR$ (4) : REM CTRL-D
3  PRINT D$; "OPEN LISTING"
4  PRINT D$; "WRITE LISTING"
5  POKE 33,30
6  LIST 2270,5130
7  PRINT D$; "CLOSE LISTING"
8  TEXT : END
```

Les numéros de ligne de ce programme sont séquentiels afin de pouvoir ajouter ce programme existant déjà en mémoire. On peut aussi le placer à la fin d'un programme existant en mémoire.

Le programme CAPTURE crée un fichier séquentiel en mode texte, contenant les instructions BASIC précédées de leur numéro. Si ce fichier est utilisé avec une commande EXEC,

```
EXEC CAPTURE
```

tout se passera comme si les instructions BASIC du programme étaient une à une tapées du clavier, et enregistrées dans la mémoire du micro-ordinateur. Une fois enregistré dans le fichier, le programme peut être modifié, puis réenregistré dans la mémoire à l'aide de la commande EXEC. Contrairement aux commandes LOAD et RUN, EXEC n'efface pas un programme qui se trouvait précédemment en mémoire.

Utilisant le programme CAPTURE, vous « capturez » un programme écrit dans un langage, le BASIC simple, par exemple, puis vous le renvoyez en mémoire par la commande EXEC, dans un autre langage, le BASIC étendu, par exemple. bien entendu, le nouveau programme nécessitera quelques modifications car la syntaxe est parfois différente d'un BASIC à l'autre.

On peut aussi utiliser EXEC de cette manière pour ajouter de nouvelles lignes dans un programme existant en mémoire. En fait, on peut aussi sauvegarder une liste du programme CAPTURE dans un fichier en mode texte puis, à l'aide d'une commande EXEC, insérer CAPTURE dans un programme, à chaque fois qu'on le désire.

La conversion de routines langage machine en BASIC

Voici un autre programme très utile qui permet de convertir une routine écrite en langage machine en une routine écrite en langage BASIC, à l'aide de l'instruction POKE. Ce programme peut être inséré dans un programme BASIC simple, ou dans un programme BASIC étendu, pour mettre la routine en langage machine à chaque fois que le programme BASIC est exécuté :

```
  5  REM CODE-POKES WRITER
 10  D$ = "" : REM CTRL-D
 15  PRINT D$; "OPEN CODE-POKES"
 20  PRINT D$; "DELETE CODE-POKES"
 25  PRINT D$; "OPEN CODE-POKES"
 30  PRINT D$; "WRITE CODE-POKES"
 40  LINENUMBER = 7000
 50  FOR PLACE = 768 TO 783
 60  COUNTER = COUNTER + 1
 70  IF COUNTER = 10 THEN COUNTER = 1
 80  IF COUNTER <> 1 THEN 120
 90  PRINT
100  PRINT LINENUMBER;
110  LINENUMBER = LINENUMBER + 1
120  PRINT "POKE"; PLACE; ","; PEEK (PLACE); ":";
130  NEXT PLACE
135  PRINT
140  PRINT D$; "CLOSE CODE-POKES"
150  END
```

Lorsque ce programme est utilisé, le nombre donné en ligne 40 (dans cet exemple, 7000) doit être modifié, et remplacé par le numéro de ligne dans le programme BASIC où les POKE correspondants à la routine doivent commencer. La boucle FOR qui commence en ligne 50 doit avoir comme valeur de départ et comme valeur limite l'adresse mémoire (en position décimale) de la routine en langage machine destiné à être convertie.

Après avoir tapé le programme, exécutez-le. Le fichier CODE-POKES sera créé. Puis utilisez la commande EXEC :

```
EXEC CODE-POKES
```

pour placer votre routine en langage machine dans n'importe quel programme BASIC (simple ou étendu), en commençant au numéro de ligne spécifié.

La commande MAXFILES et le BASIC simple

Un fichier EXEC doit être utilisé si l'on désire augmenter le nombre de mémoires tampon par MAXFILES, à l'intérieur d'un programme en BASIC simple, sans effacer le programme. Voici comment. Il faut utiliser la méthode décrite plus haut pour créer un fichier de procédure cataloguée. Appelons-le FILE.EX.

Le fichier FILE.EX. doit contenir une commande fixant la mémoire haute (HIMEM) à une adresse inférieure à celle utilisée par les mémoires tampon supplémentaires définies par MAXFILES (595 octets sont pris par chaque mémoire tampon supplémentaire).

Par exemple, FILE.EX. doit contenir les commandes suivantes pour permettre la réservation de 5 mémoires tampon :

```
MAXFILES 5
LOAD PROGRAM
DEL 10, 20
RUN
```

Les premières lignes du programme doivent être comme montrées ci-dessous. Le CTRL-D doit être tapé directement, aucune variable ne pouvant être définie avant ces instructions.

```
10  PRINT "CTRL-D EXEC FILE.EX"
20  END
30  REM MAIN PROGRAM
```

La commande EXEC

La syntaxe est :

```
EXEC f
```

où f est le nom d'un fichier séquentiel en mode texte contenant des commandes BASIC ou des lignes de programme. Des exemples d'utilisation ont été développés au début de ce chapitre.

Lorsqu'une commmande EXEC s'exécute, la première rubrique du fichier appelé est chargée en mémoire, comme si elle venait d'être tapée au clavier. Puis le système tente d'exécuter l'ordre s'il s'agit d'une commande BASIC, ou l'enregistrer s'il s'agit d'une ligne de programme BASIC. Le type de BASIC (simple ou étendu) n'est pas modifié par la commande EXEC, sauf si le fichier contient une commande FP ou INT. Lorsque l'exécution est terminée pour la première rubrique du fichier, la seconde est alimentée, et ainsi de suite jusqu'à ce que la dernière rubrique du fichier ait été traitée.

La commande EXEC *ne peut pas être interrompue* par un CTRL-C.

Un seul fichier de procédure cataloguée peut être ouvert à un moment donné. Si un fichier de procédure cataloguée contient lui-même une commande EXEC, il sera fermé avant que cette nouvelle commande EXEC ne soit exécutée. Si un fichier de procédure cataloguée contient une commande RUN d'un programme, la procédure attendra patiemment la fin du programme, avant de reprendre la main et de passer à l'enregistrement suivant du fichier.

Attention : lorsqu'un programme est exécuté à partir d'une procédure cataloguée, *il ne doit pas comporter d'instruction INPUT,* faute de quoi l'INPUT ne porterait pas sur le clavier, mais sur le prochain enregistrement du fichier de procédure. De plus, si le prochain enregistrement du fichier de procédure comporte une commande directe du DOS, cette commande sera exécutée alors que le programme est en cours d'exécution. Les résultats peuvent être déroutants.

Si une interruption par CTRL-C intervient au cours de l'exécution d'un programme écrit en BASIC étendu, alors qu'un fichier de procédure est ouvert, le reste de la procédure sera généralement ignoré.

Si une rubrique d'un fichier de procédure ne peut pas être interprétée, soit en tant que ligne de programme, soit en tant que commande, le message

SYNTAX ERROR

sera affiché, et la prochaine rubrique du fichier sera alimentée dans le micro-ordinateur. On peut donc utiliser cette méthode, assez rudimentaire, pour examiner le contenu de n'importe quel fichier écrit en mode texte (à condition d'être en mode MON C,I,O). Il convient de ne pas oublier de sauvegarder éventuellement au préalable le programme que l'on avait en mémoire.

La commande EXEC peut aussi être utilisée avec un paramètre R de position relative de rubrique, d'une manière quelque peu différente de l'utilisation par la commande POSITION de ce même paramètre.

La syntaxe pour cette utilisation est :

EXEC f (,Rp)

où Rp indique que la première procédure à exécuter dans le fichier f est située dans la nième rubrique. Étant donné que la commande EXEC initialise le pointeur de fichier au premier caractère du fichier, le paramètre Rp donne la pième rubrique, à partir du *début du fichier.* p correspond donc à la rubrique réelle. RØ fait commencer à la première procédure du fichier, R1 à la seconde, et ainsi de suite.

Remarquez qu'il s'agit d'une différence fondamentale avec l'utilisation du paramètre R dans la commande POSITION, où R3, par exemple, est un numéro *relatif de rubrique,* pouvant désigner des rubriques différentes à différents moments de l'exécution du programme.

EXEC MONFICHIER, Rf

génère le message d'erreur END OF DATA si le paramètre R spécifie une rubrique qui est située à partir de la seconde rubrique après la fin du fichier (aucun message n'est affiché si R spécifie la rubrique située immédiatement après la fin du fichier, mais rien ne se produit).

CHAPITRE 8
Les fichiers à accès direct

78 Le fonctionnement des fichiers à accès direct
78 L'accès à un enregistrement déterminé
80 Plusieurs enregistrements
81 Un programme de démonstration de l'accès direct
83 L'écriture et la lecture des fichiers en accès direct

Le fonctionnement des fichiers à accès direct

Les fichiers à accès direct sont organisés comme un ensemble de cellules de taille équivalente dans un rayon de miel : certaines cellules sont remplies, d'autres ne le sont pas nécessairement. L'équivalent de chaque cellule est nommé un *enregistrement*. Lorsqu'un fichier à accès direct est créé, on doit spécifier la longueur standard des enregistrements que le fichier contiendra par la suite.

Contrairement aux rubriques d'un fichier séquentiel qui peuvent être de longueur variable, les enregistrements d'un fichier à accès direct ont une *longueur fixe déterminée*. A la première écriture d'un enregistrement dans le fichier, une place suffisante est réservée pour un enregistrement de la taille standard, qu'il soit rempli ou non. Les fichiers à accès direct ne sont pas nécessairement la meilleure utilisation possible de la place disponible sur disquette. Par contre, l'organisation régulière des informations enregistrées en facilite la consultation.

Une organisation permettant l'accès direct doit être adoptée dans le cas où les applications nécessitent un accès rapide à différentes informations du fichier, ou bien lorsque certains enregistrements du fichier doivent être mis à jour souvent et rapidement. Par exemple, un fichier à accès direct est particulièrement adapté pour une tenue de stock en temps réel.

Les fichiers à accès direct sont créés et lus d'une manière très semblable aux fichiers séquentiels. La différence essentielle est que certaines commandes ont des paramètres complémentaires : la commande OPEN comporte un paramètre L qui représente la longueur standard d'un enregistrement, alors que les commandes READ et WRITE comportent un paramètre R qui représente le numéro d'enregistrement.

Quelques exemples d'utilisation seront donnés avant de voir en détail comment on écrit et comment on lit un fichier à accès direct. Une information plus technique est donnée en annexe C.

L'accès à un enregistrement déterminé

Comment peut-on accéder à un enregistrement déterminé? Les deux programmes suivants, écrits en BASIC étendu, illustrent comment le DOS peut le faire.

Le premier programme saisit un nom (N$), un numéro de téléphone (P$) et un code postal (Z$), puis les écrit dans le premier enregistrement d'un fichier nommé COURRIER :

```
10  REM ÉCRITURE DE COURRIER
20  D$ = CHR$ (4) : REM CTRL-D
30  INPUT "NOM:            "; N$
40  INPUT "TEL.:           "; P$
50  INPUT "CODE POST:   "; Z$
60  PRINT D$; "OPEN COURRIER, L200"
70  PRINT D$; "WRITE COURRIER, R1"
80  PRINT N$ : PRINT P$ : PRINT Z$
90  PRINT D$; "CLOSE COURRIER"
```

La ligne 20 place un CTRL-D dans la variable D$, comme d'habitude. De la ligne 30 à la ligne 50, l'information à enregistrer est acceptée à partir du clavier. Il ne faut pas taper de virgule ou de point-virgule dans ces informations.
En ligne 60, un fichier nommé COURRIER est ouvert.
La ligne 70 annonce l'écriture de l'enregistrement 1 du fichier COURRIER.
La ligne 80 dirige le contenu de N$, P$, et de Z$ vers la disquette. Le contenu de l'ensemble des trois variables, séparées par des retours chariot, est mémorisé dans l'enregistrement 1.
En ligne 90, le fichier COURRIER est clôturé.

Lors de l'exécution, sous l'effet de MON C,I,O, vous verrez, par exemple :

```
NOM :          MICHEL FOURNIER
TEL. :         187 46 55
CODE POST : 75017
OPEN COURRIER, L200
WRITE COURRIER, R1
MICHEL FOURNIERE
187 46 55
75017
CLOSE COURRIER
```

L'enregistrement numéro 1 du fichier COURRIER peut être lu par le programme suivant :

```
10  REM LECTURE DE COURRIER : A
20  D$ = CHR$ (4) : REM CTRL-D
30  PRINT D$; "OPEN COURRIER, L200"
40  PRINT D$; "READ COURRIER, R1"
50  INPUT N1$,P1$,Z1$
70  PRINT D$; "CLOSE COURRIER"
```

Lors de l'exécution, le programme donnera l'affichage suivant (comme d'habitude, la paire de points d'interrogation est causée par une instruction INPUT avec plus d'une réponse) :

```
OPEN COURRIER, L200
READ COURRIER, R1
?MICHEL FOURNIERE
??187 46 55
??75017
CLOSE COURRIER
```

Et voici un autre programme légèrement différent qui peut aussi lire l'enregistrement 1 du fichier COURRIER :

```
10  REM LECTURE DE COURRIER : B
20  D$ = "" : REM CTRL-D
30  PRINT D$; "OPEN COURRIER, L200"
40  PRINT D$; "READ COURRIER, R1"
50  INPUT N1$
60  INPUT P1$
70  INPUT Z1$
80  PRINT D$; "CLOSE COURRIER"
90  END
```

Plusieurs enregistrements

Le programme que nous venons de réaliser écrivait un seul enregistrement, constitué de trois rubriques, dans le fichier à accès direct COURRIER.

Le prochain exemple illustre les possibilités d'écrire plusieurs enregistrements dans un fichier nommé RA-FILE. Dans cet exemple, ce sont les enregistrements de numéro 12 à 15 qui sont écrits :

```
  5  REM ÉCRITURE DE RA-FILE
 10  D$ = CHR$ (4) : CTRL-D
 20  PRINT D$; "OPEN RA-FILE"
 30  PRINT D$; "DELETE RA-FILE"
 40  PRINT D$; "OPEN RA-FILE, L30"
 50  FOR I = 12 TO 15
 60  PRINT D$; "WRITE RA-FILE, R"; I
 70  PRINT "NOM ADRESSE"; I
 80  NEXT I
 90  PRINT D$; "WRITE RA-FILE, R13"
100  PRINT "DOS"
110  PRINT D$; "CLOSE RA-FILE"
```

La ligne 10 initialise la variable D$ avec CTRL-D.
Les lignes 20 et 30 permettent de s'assurer que le fichier RA-FILE ne contient pas déjà des informations.
En ligne 40, le fichier RA-FILE, dont les enregistrements ont une longueur de 30 octets, est ouvert.
De la ligne 50 à la ligne 80, une boucle FOR permet d'écrire quatre fois la même information : NOM ADRESSE.
En ligne 60, la valeur du paramètre R, numéro d'enregistrement, est donnée par l'indice I de la bocle FOR, qui progresse de 12 à 15.
De la ligne 90 à la ligne 100, l'enregistrement de numémro 13 est à nouveau écrit, mais cette fois, avec une information différente.
En ligne 110, le fichier RA-FILE est clôturé.

L'affichage sera comme suit, lors de l'exécution, sous l'effet de MON C,I,O :

```
OPEN RA-FILE
DELETE RA-FILE
OPEN RA-FILE, L30
WRITE RA-FILE, R12
NOM ADRESSE 12
WRITE RA-FILE, R13
NOM ADRESSE 13
WRITE RA-FILE, R14
NOM ADRESSE 14
WRITE RA-FILE, R15
NOM ADRESSE 15
WRITE RA-FILE, R13
DOS
CLOSE RA-FILE
```

De la même manière, on peut lire un ou plusieurs enregistrements déterminés d'un fichier à accès direct. Le prochain programme lit les enregistrements 12 à 15 du fichier RA-FILE, et recherche (en ligne 60) le ou les enregistrements qui contiennent les lettres DOS comme trois premiers caractères :

```
 5  REM LECTURE DE RA-FILE
10  D$ = CHR$ (4) : REM CTRL-D
20  PRINT D$; "OPEN RA-FILE, L30"
30  FOR J = 12 TO 15
40  PRINT D$; "READ RA-FILE, R"; J
50  INPUT AS
60  IF LEFT$ (A$,3) = "DOS" THEN PRINT "L'ENREGISTREMENT→"; J; "→A ÉTÉ
MODIFIÉ"
70  NEXT J
80  PRINT D$; "CLOSE RA-FILE"
```

La ligne 10 initialise la variable D$ à la valeur CTRL-D.
La ligne 20 ouvre le fichier RA-FILE, dont la longueur des enregistrements est de 30 octets.
De la ligne 30 à la ligne 70, une boucle FOR est exécutée pour explorer le fichier RA-FILE, du 12e enregistrement jusqu'au 15e.
En ligne 60, les trois premiers caractères de l'enregistrement qui vient d'être lu sont comparés aux lettres DOS. Si l'égalité est trouvée, le message « L'ENREGISTREMENT r A ÉTÉ MODIFIÉ », r étant une valeur de J au moment de l'égalité, s'affiche.
En ligne 80, le fichier RA-FILE est clôturé.

L'affichage suivant apparaîtra à l'écran lors de l'exécution sous l'effet de MON C,I,O :

```
OPEN RA-FILE, L30
READ RA-FILE, R12
?NOM ADRESSE 12
READ RA-FILE, R13
?DOS
L'ENREGISTREMENT 13 A ÉTÉ MODIFIÉ
READ RA-FILE, R14
?NOM ADRESSE 14
READ RA-FILE, R15
? NOM ADRESSE 15
CLOSE RA-FILE
```

Remarquez, dans ce dernier programme, que l'on ne tente de lire que les enregistrements qui ont été préalablement écrits. En effet, si l'on avait tenté, par exemple, de lire l'enregistrement 8 de RA-FILE, le message END OF DATA aurait été affiché, puisqu'aucune information n'avait été enregistrée à cet endroit.

De même, si plus d'une instruction INPUT avait été lancée pour l'un des enregistrements, le même message aurait été affiché : chaque enregistrement, dans ce cas particulier, ne contient qu'une seule rubrique.

Un programme de démonstration de l'accès direct

Le dernier, et non le moindre, des programmes de démonstration enregistrée sur la disquette Maître du système se nomme RANDOM, et illustre une petite tenue de stock. Petite, en effet, car le nombre maximum d'articles dans le fichier est 9. Cette limite est fixée afin que le programme reste très simple. Mais il est bien évident que le micro-ordinateur est capable de traiter des milliers d'articles dans une tenue de stock.

Au début de l'exécution, le programme se copie ainsi que le fichier à accès direct, sur une disquette non protégée contre l'écriture, puis il s'exécute. Vous pouvez afficher l'un ou la totalité des articles.

Voici comment procéder :

1. A partir de la disquette Maître du système, tapez

   RUN RANDOM

   le message suivant apparaîtra :

   THIS DEMONSTRATION WILL NOT EXECUTE ON A WRITE-PROTECTED
   DISKETTE SUCH AS YOUR DOS SYSTEM MASTER
   FOR YOUR CONVENIENCE, PROVISIONS HAVE BEEN MADE TO COPY
   THIS PROGRAM AND IT'S DATA TO ANOTHER DISKETTE.

   DO YOU WISH TO DO THIS NOW? (Y OR N)

Ce qui signifie :

Cette démonstration ne peut pas être exécutée avec une disquette protégée contre l'écriture telle que votre Système Maître DOS.

Pour votre commodité, il est prévu de copier ce programme et son fichier sur une autre disquette. Désirez-vous le faire maintenant? (O ou N)

Si vous tapez N (Non), vous revenez simplement au BASIC étendu.

2. Si vous tapez Y (Oui), le message suivant sera affiché :

   NOW READING DATA...

   ce qui signifie : la lecture d'informations est en cours.

   Puis le message

   INSERT AN INITIALIZED DISKETTE, THEN
   PRESS THE RETURN KEY TO BEGIN TRANSFER

   sera affiché. Il signifie : insérez une disquette initialisée et frappez la touche RETURN pour commencer le transfert.

3. Retirez la disquette Maître du système, placez une disquette non protégée contre l'écriture dans le lecteur, et frappez la touche RETURN. L'exécution du programme commence après l'affichage d'un message d'attente.

4. L'affichage suivant apparaîtra à l'écran :

COMMAND	APPLE PROMS NUMBER
LIST	1
CHANGE	2
EXIT	3
CHOOSE NUMBER (1 - 3)	

   Tapez 1, et vous verrez l'affichage

PART NUMBER 1 - 9 (Ø = ALL) Ø

ce qui signifie : numéro d'article 1 - 9 (Ø pour tous les articles).

5. Tapez Ø, pour obtenir la liste de tous les articles dans ce système de tenue de stock. Vous verrez :

PART#	NAME	SIZE	IN STOCK
1	PARALLEL PRINT	256	5ØØ
2	COMMUNICATIONS	256	125Ø
3	(NOT AVAILABLE)	256	Ø
4	(NOT AVAILABLE)	256	Ø
5	DISK BOOT	256	432
6	STATE MACHINE	256	46Ø
7	SERIAL PRINTER 1	256	878
8	SERIAL PRINTER 2	512	741
9	CENTRONICS	256	129Ø

PRESS THE RETURN KEY TO CONTINUE

Lorsque vous désirez retourner à la page des options, frapper la touche RETURN.

6. Essayez les différentes options du programme. Le choix numéro 1 vous permet d'afficher les articles un à la fois, ou tous en même temps.

Le choix numéro 2 vous permet de changer les noms et descriptions des articles, un par un, ou tous à la fois. Supposons, par exemple, que l'article 3 devrait être APPLE II, taille 48K, avec un stock de 1234. Voici comment modifier l'article 3 :
- choisir l'option 2, CHANGE;
- choisir l'article numéro 3;
- le nom précédent est affiché, le curseur clignotant étant positionné sur le premier caractère, de manière à vous permettre de le remplacer par le nouveau. Lorsque vous frapperez la touche RETURN, le curseur se placera sur la taille, puis sur la quantité, pour vous permettre d'enregistrer les nouvelles informations.

Pour conserver les informations anciennes, sans les modifier, il suffit de ne pas en taper de nouvelles, mais de frapper simplement la touche RETURN.

Le choix 3 termine l'exécution du programme.

L'écriture et la lecture des fichiers en accès direct

L'utilisation de la commande CLOSE est identique à celle développée pour les fichiers à accès séquentiel. Par contre, la syntaxe pour la commande OPEN comporte un paramètre supplémentaire *qui doit être obligatoirement spécifié.*

OPEN f Lj (,Ss) (,Dd) (,Vv)

La lettre L est l'abréviation de la longueur de l'enregistrement; le nombre j représente le nombre d'octets (caractères ou chiffres) qui doit être alloué pour chaque enregistrement du fichier à accès direct concerné. Si l'option L est omise, la valeur par défaut de j est 1. Le nombre J doit être compris entre 1 et 32767.

Supposons que lors d'une commande OPEN précédant une commande READ, la longueur spécifiée pour le fichier soit différente de celle qui avait été spécifiée lors de l'écriture, le DOS suivra aveuglément la nouvelle indication, pour calculer la position du pointeur dans le fichier.

Il faut donc conserver une documentation écrite sur la structure et le contenu des différents fichiers — certains programmeurs mémorisent cette information dans l'enregistrement Ø du fichier. Il peut être aussi utile d'inclure la longueur des enregistrements du fichier dans le nom même du fichier :

```
RANDFILES:L20
STOCKLISTS-L100
DIRECTORIES(L50)
```

Il n'y a pas moyen, en partant exclusivement du fichier, de détermier la longueur des enregistrements. Il faut donc que cette information fasse partie de la documentation.

Les enregistrements ne doivent jamais être plus longs que le nombre d'octets spécifié dans le paramètre L, faute de quoi des enregistrements pourraient comporter des recouvrements, et le fichier deviendrait inutilisable.

Les commandes READ et WRITE comportent un paramètre R qui doit être utilisé pour lire et pour écrire un enregistrement déterminé dans un fichier à accès direct.

```
WRITE f (,Rr)
READ f (,Rr)
```

Exemples : WRITE LISIBLEMENT, R3
READ VITE, R13

Le paramètre Rr (Record = enregistrement) est utilisé pour créer (avec WRITE) ou pour consulter (avec READ) le rième enregistrement du fichier. La valeur par défaut de r est Ø, désignant le premier enregistrement du fichier.

L'utilisation du CTRL-C pour interrompre une commande READ en BASIC étendu engendre un message d'erreur :

```
REENTER
```

Pour l'éviter, interrompre avec RESET au lieu de CTRL-C.

On peut considérer, d'une certaine manière, que chaque enregistrement d'un fichier à accès direct est en même temps un court fichier séquentiel. Les commandes WRITE et READ peuvent être utilisées avec un paramètre Byte (octet), en plus du paramètre R. Le paramètre B spécifie l'octet, à l'intérieur de l'enregistrement, à partir duquel le PRINT (après le WRITE) ou le INPUT ou GET (après READ) débutera.

```
WRITE f (,Rr) (,Bb)
READ f (,Rr) (,Bb)
```

Si le paramètre B est utilisé, l'écriture, ou la lecture, débute au bième octet de l'enregistrement. L'utilisation du paramètre B nécessite évidemment une connaissance parfaite du contenu des enregistrements, octet par octet.

On peut aussi utiliser la commande POSITION pour déplacer le pointeur de fichier à l'intérieur d'un enregistrement. Après qu'une commande READ ou WRITE ait été donnée, on peut utiliser POSITION pour faire avancer le pointeur de fichier. Cependant, la commande POSITION annule une commande READ ou une commande WRITE. Il est donc nécessaire de donner à nouveau une commande READ ou WRITE, cette fois sans paramètre R, avant de pouvoir effectivement lire le fichier, ou y écrire des informations.

Voir l'annexe C pour plus de détails sur la manière dont les informations sont mémorisées sur disquette, dans les fichiers à accès direct, notamment.

CHAPITRE 9
Le mode binaire

88 Les fichiers en langage machine
88 BSAVE
89 BLOAD
89 BRUN
89 Les routines RWTS

Les fichiers en langage machine

Le DOS permet de sauvegarder sur disquette des informations contenues dans la mémoire du micro-ordinateur, ou de les lire à partir de la disquette.

Nous avons déjà vus les commandes DOS SAVE, LOAD et RUN : ces commandes concernent un contenu de mémoire interprété, comme les instructions d'un programme BASIC le sont. Par contre, les commandes BSAVE, BLOAD et BRUN que nous allons voir maintenant portent sur n'importe quel contenu de mémoire, sous sa forme la plus élémentaire, à savoir sa forme binaire (qui peut être exprimée également en notation hexadécimale).

Le B qui a été ajouté devant chaque commande signifie Binaire; les noms des fichiers dont les informations sont mémorisées sous forme binaire sont aussi précédés de la lettre B, dans le catalogue.

Un fichier binaire est une image exacte, bit à bit, d'une *portion de la mémoire* du micro-ordinateur, définie par des adresses. A l'intérieur des limites données par les adresses, il peut aussi bien y avoir :
–un programme en langage machine,
– des informations, sous forme binaire,
– une image bit à bit d'un écran en graphique haute résolution.

BSAVE

La commande BSAVE sauvegarde un fichier f représentant le contenu d'un segment de mémoire. La syntaxe est :

BSAVE f ,Aa,Lj (,Ss) (,Dd) (,Vv)

où les paramètres habituels S, D et V représentent respectivement le numéro de port, le numéro de lecteur et le numéro de volume. Remarquez que les paramètres A et L ne sont pas optionnels.
Le paramètre A spécifie l'adresse de début (soit en notation décimale, soit en notation hexadécimale) de la portion à écrire sur disquette. Si l'adresse est exprimée en notation hexadécimale, le signe $ doit la précéder. Si la valeur du paramètre A est inférieure à Ø, ou plus grande que 65535, un message d'erreur SYNTAX ERROR est affiché.

On ne peut donc pas utiliser des adresses négatives équivalentes avec cette commande.

A l'intérieur des limites Ø - 65535, aucun message d'erreur ne sera engendré si l'adresse de début indiquée est en dehors de la limite de la mémoire utilisateur dont le micro-ordinateur est équipé. Le paramètre A ne doit pas avoir un contenu supérieur à l'adresse maximum de mémoire (49151 en décimal, ou $BFFF, sur un système 48K).

Le paramètre L spécifie la longueur, en nombre d'octets, du segment mémoire à sauvegarder. Si le paramètre L comporte une valeur inférieure à Ø, ou supérieure à 65535, le message SYNTAX ERROR sera affiché.

Le nombre maximum d'octets pouvant être sauvegardés dans un fichier binaire sur disquette est 32767. Si l'on a besoin d'en sauvegarder plus, il suffit d'utiliser deux fois BSAVE. A l'intérieur des limites Ø - 32767, aucun message d'erreur n'est affiché si le paramètre L donne une adresse qui n'existe pas physiquement dans le micro-ordinateur.

L'exemple ci-dessous est la sauvegarde d'un fichier nommé PICTURE (image), contenant l'image de la mémoire réservée aux graphiques haute résolution. Cet exemple est écrit de quatre manières différentes et équivalentes, qui toutes sont identiques du point de vue opérationnels :

```
BSAVE PICTURE, A$4000, L$2000
BSAVE PICTURE, A16384, L8192
BSAVE PICTURE, A16384, L$2000
BSAVE PICTURE, A$4000, L8192
```

BLOAD

La commande BLOAD permet de charger en mémoire le contenu d'un fichier binaire.

La commande BLOAD n'efface pas un programme BASIC qui se trouverait éventuellement en mémoire, sauf si les adresses spécifiées dans le BLOAD concernent une portion de la mémoire déjà occupée par le programme BASIC.

La syntaxe est :

```
BLOAD f (,Aa) (,Ss) (,Dd) (,Vv)
```

où S, D et V sont les paramètres habituels.

Si le paramètre A est donné, le chargement est effectué à partir de l'adresse a. Si le paramètre A n'est pas donné, l'adresse de chargement est l'adresse d'origine, à partir de laquelle le BSAVE a été exécuté.

Supposons que le fichier PICTURE contienne l'image mémoire d'un graphique haute résolution. L'exemple suivant charge cette image dans la première zone haute résolution de la mémoire du micro-ordinateur. Les deux écritures différentes dans la forme sont identiques du point de vue opérationnel :

```
BLOAD PICTURE, A8192
BLOAD PICTURE, A$2000
```

BRUN

La syntaxe de cette commande est identique à celle de BLOAD :

```
BRUN f (,Aa) (,Ss) (,Dd) (,Vv)
```

Le fichier binaire f est obligatoirement un programme écrit en langage machine.

En premier lieu, la commande BRUN exécute un BLOAD (le paramètre A a donc la même fonction que la commande BLOAD). Après le chargement, l'exécution se positionne à l'adresse a par l'instruction JMP (langage machine). Si le programme est bien un programme en langage machine, il s'exécute.

La routine RWTS

Normalement, c'est au moyen du DOS que l'on peut utiliser les disquettes.

Il existe cependant une autre méthode qui peut être employée à partir d'un programme en langage machine. Vous pouvez passer ce chapitre si vous ne maîtrisez pas le langage machine.

Pour accéder directement à la disquette, on peut utiliser les routines RWTS, qui font partie du DOS. Les lettres RWTS sont l'abréviation de Read or Write a Track and Sector, ce qui signifie lire ou écrire une piste et un secteur. Dans les explications qui suivent, tous les nombres qui sont précédés du signe $ sont en notation hexadécimale.

Chaque disquette initialisée sur le secteur DISK II comporte 35 pistes, numérotées de Ø à 34. Ces pistes peuvent être considérées comme le sillon d'un disque de musique, sauf que le sillon est continu, alors que les pistes ne le sont pas. En fait, les pistes sont des cercles concentriques, le large trou au centre de la disquette étant leur centre commun. La piste Ø est celle qui est sur le bord de la disquette, alors que la piste 34 est celle qui est la plus rapprochée du centre.

La tête de lecture est magnétique, et se déplace d'une piste à l'autre, soit pour lire, soit pour écrire des informations.

Chaque piste de la disquette comporte 16 secteurs. Les secteurs sont des groupes d'octets pré-déterminés; grâce aux secteurs, l'utilisateur peut manipuler des blocs de 256 octets, au lieu de devoir manipuler les 4Ø96 octets de la piste. Les secteurs sont numérotés, à l'intérieur d'une piste, de Ø à 15. Lorsque la disquette tourne, chaque secteur d'une piste déterminée passe sous la tête de lecture, permettant la lecture ou l'écriture d'un secteur donné.

Chaque secteur comporte deux parties : la zone adresse, et la zone données. La zone adresse comporte plusieurs indications : le numéro de la piste sur laquelle la tête est située, le numéro du secteur et le numéro de volume de la disquette. La zone donnée comporte les 256 octets d'informations mémorisées sur ce secteur sous une forme cryptée.

La routine RWTS permet à l'utilisateur d'écrire ou de lire directement une piste et un secteur sur disquette, en langage machine. Pour utiliser les routines RWTS, on doit au préalable créer une table IOB (Input/Output control Blocks), et une « Table des Caractéristiques d'Unité » correspondante.

La table IOB indique aux routines RWTS :
- le numéro de port de la disquette,
- le numéro de lecteur,
- le numéro de volume,
- la piste et le secteur concernés,
- s'il s'agit d'une lecture ou d'une écriture,
  dans quelle zone mémoire lire ou écrire.

La « Table des Caractéristiques d'Unité » fournit aux routines RWTS des informations indispensables pour opérer sur les secteurs DISK II.

Les tables IOB et caractéristiques doivent donc être mises quelque part en mémoire. Un petit programme de contrôle doit aussi être écrit et chargé en mémoire. Ce petit programme doit se débrancher, par l'instruction JSR, à l'adresse de début des routines RWTS (c'est-à-dire à l'adresse $3D9). Au début du débranchement aux routines, les registres A et Y doivent contenir l'adresse de début de la table IOB. Le registre A contiendra l'octet le plus significatif de l'adresse, et le registre Y contiendra l'octet le moins significatif. Les formats des tables IOB et Caractéristiques sont donnés à la fin de ce chapitre.

Voici un exemple d'utilisation des routines RWTS. La table IOB, donnée à titre d'exemple, la table des caractéristiques ainsi qu'un petit programme de contrôle seront chargés en mémoire à partir de l'ordre $CØØ.

Le petit programme de contrôle suivant charge les registres A et Y avec l'adresse de début de la table IOB, et se débranche (JSR) à la routine RWTS :

$C00	A9 0C	LDA	#$0C	chargement du registre A avec $0C
$C02	A0 0A	LDY	#$0A	chargement du registre Y avec $0A
$C04	20 D9 03	JSR	$03D9	saut aux routines RWTS
$C07	60	RTS		
$C08	00	BRK		

La table IOB qui suit serait utilisée pour écrire 256 octets de mémoire, à partir de l'adresse de début (mémoire tampon) $2000, sur le secteur 6 de la piste 18 de la disquette insérée dans le lecteur 1 du port 6.

ADRESSE	CODE	OBJET
$C0A	01	IOB type indiqué, doit être $01
$C0B	60	Numéro de port, multiplié par 16
$C0C	01	Numéro de lecteur
$C0D	00	Numéro de volume attendu
$C0E	12	Numéro de piste
$C0F	06	Numéro de secteur
$C10	20	Octet le moins significatif de l'adresse de la table des caractéristiques de l'unité
$C11	0C	Octet le plus significatif de l'adresse de la table des caractéristiques de l'unité
$C12	00	Octet le moins significatif de l'adresse de début de la mémoire tampon
$C13	20	Octet le plus significatif de l'adresse de début de la mémoire tampon
$C14	00	Inutilisé
$C15	00	Inutilisé
$C16	02	Code commande : $02 = WRITE
$C17	00	Code d'erreur
$C18	00	Numéro de volume effectif
$C19	60	Précédent numéro de port x 16 utilisé
$C1A	01	Précédent lecteur utilisé

La table des caractéristiques de l'unité suivante doit être entrée. Nous la placerons à l'adresse $C20, juste après la table IOB. Les adresses $C10 et $C11 de la table IOB donnent l'adresse de début de la table des caractéristiques de l'unité.

ADRESSE	CODE	OBJET
$C20	00	Type d'unité ($00 pour DISK II)
$C21	01	Nombre de phases par piste (01 pour DISK II)
$C22	EF	Compteur de temps (mettre $EF)
$C23	D8	Compteur de temps (mettre $D8)

Lorsque le chargement de la table IOB à partir de $C0A, le chargement de la table des caractéristiques de l'unité à partir de $C20, et le chargement du programme de contrôle à partir de $C00 sont réalisés, vos pouvez taper
soit C00G
soit CALL 3072
pour exécuter l'ensemble.

TABLE 3: FORMAT DE LA TABLE IOB

OCTET	NOM	OBJET
1	IBTYPE	Indique aux routines RWTS de quel type de table IOB il s'agit. Doit être $01. Aucun autre code n'est actuellement défini.
2	IBSLOT	Doit contenir le numéro de port x16, dans lequel est inséré l'interface de la disquette. Par exemple, pour le port#6, la valeur $60 doit être mise ici.

3	IBDRVN	Doit contenir le numéro de lecteur concerné. Ne peut être que Ø1 ou Ø2.
4	IBVOL	Doit contenir le numéro de volume de la disquette concernée. Si $ØØ, n'importe quel numéro de volume sera accepté.
5	IBTRK	Doit contenir le numéro de piste auquel on désire accéder (Ø à 34). Doit être compris dans les limites $ØØ à $22.
6	IBSECT	Doit contenir le numéro de secteur auquel on désire accéder (Ø à 15). Doit être compris dans les limites $ØØ à $ØF.
7 & 8	IBDCTP	Ces deux octets contiennent l'adresse de début de la Table des Caractéristiques d'Unité. L'octet 7 contient l'octet de faible poids de l'adresse, et l'octet 8 l'octet fort poids.
9 & 1Ø	IBBUFP	L'octet 9 et l'octet 1Ø contiennent l'adresse de début de la mémoire tampon (où les données sont stockées). La mémoire tampon est obligatoirement une zone de 256 octets de longueur, sur laquelle opéreront les routines RWTS. S'il s'agit d'une écriture, l'information contenue dans les 256 octets de la mémoire tampon sera écrite sur disquette. S'il s'agit d'une lecture, l'information en provenance de la disquette sera enregistrée dans les 256 octets de la mémoire tampon. 256 octets, c'est à la fois le minimum et le maximum du nombre d'octets qui peuvent être lus, ou écrits, en une opération.
11 & 12		Inutilisés.
13	IBCMD	Cet octet contient le code commande des routines RWTS. Les valeurs qui peuvent être données sont : $ØØ : commande nulle. Met la disquette en marche et positionne la tête de lecture/écriture. Ne fait rien d'autre. $Ø1 : lecture des 256 octets mémorisés sur la disquette, aux piste/secteur spécifiés, et chargement des informations dans la zone tampon définie. $Ø2 : écriture des 256 octets de la mémoire tampon sur la disquette, aux piste/secteur spécifiés. $Ø4 : formatage de la disquette. Lorsqu'une disquette est formatée, des bits de synchronisation sont écrits sur chaque piste et sur chaque secteur de la disquette. La totalité de la disquette étant formatée, les valeurs contenues dans les octets 5 et 6 sont ignorées dans ce cas. La totalité d'une disquette formatée est disponible pour l'utilisation. Elle ne contient pas le DOS, ni quoi que ce soit d'autre, jusqu'à ce que l'utilisateur y enregistre des informations.
14	IBSTAT	Cette mémoire contiendra un code erreur si une situation anormale a été rencontrée au cours de l'exécution. Si, à la fin de l'exécution des routines RWTS, le bit 'carry' report du registre P est égal à 1, c'est qu'une anomalie a été détectée. Cet octet indique la nature de l'erreur : S1Ø : la disquette est protégée contre l'écriture, donc l'écriture ne peut pas s'effectuer. $2Ø : le numéro de volume de la disquette ne concorde pas avec celui indiqué dans l'octet 4. $4Ø : erreur concernant le lecteur. Quelque chose d'inhabituel s'est produit. $8Ø : erreur de lecture. Les routines RWTS n'ont pas réussi, après 48 essais, à lire soit la zone adresse, soit la zone donnée du secteur. Une des causes fréquentes de cette erreur est une tentative de lecture d'un secteur où aucune information n'a été préalablement écrite.

15	IBSMOD	Le numéro de volume lu sera mémorisé ici.
16	IOBPSN	Cet octet doit contenir le numéro de port x16 qui a été le dernier utilisé. Par exemple, si la précédente commande concernait un lecteur de disquettes connecté au port 5, la valeur $50 doit être indiquée ici. S'il n'y a pas d'interface disquette dans le port indiqué, le programme sera « planté ».
17	IOBPDN	Cet octet doit contenir le numéro du lecteur de disquettes qui a été utilisé en dernier — $01 o $02.

TABLE 4 : FORMAT DE LA TABLE DES CARACTÉRISTIQUES DE L'UNITÉ

OCTET	NOM	OBJET
1	DEVTPC	Code type de l'unité — $00 pour le DISK II.
2	PPTC	$01 doit être mis ici.
3 & 4	MONTC	Décompteur de temps de rotation du moteur, en unités de 100 microsecondes. L'octet 3 doit contenir $EF, et l'octet $D8 pour les lecteurs de disquettes DISK II.

CHAPITRE 10
Les entrées/sorties et le recouvrement de programmes

96 Le choix des unités d'entrée/sortie et divers
100 La commande CHAIN en BASIC SIMPLE
101 La commande CHAIN et le BASIC ETENDU

Le choix des unités d'entrée/sortie et divers

L'unité d'entrée la plus connue et la plus courante est le clavier du micro-ordinateur. L'unité de sortie la plus connue et la plus courante est l'écran. Mais bien d'autres unités d'entrée/sortie peuvent être connectée dans un ou plusieurs des septs ports disponible, et utilisées pour l'entrée ou la sortie par l'intermédiaire des commandes IN# et PR#.

Exemples :

IN#6 transfert les opérations d'entrée postérieures à partir de l'unité dont l'interface est insérée dans le port 6. S'il n'y a pas d'interface dans le port 6, le système sera "planté".
Frappez la touche RESET pour reprendre la main.

IN#0 ramène les opérations d'entrée postérieures à partir du clavier ou bien d'un périphérique.

PR#1 transfert les opérations de sortie postérieures à l'unité contrôlée par l'interface insérée dans le port 1 (généralement l'imprimante). De même, si le port concerné ne comporte pas d'interface, le système sera « planté », et seul RESET permet de reprendre la main.

PR#0 ramène les opérations de sortie postérieures à l'écran (et non au port 0).

La syntaxe pour cette commande est :

IN#s ou PR#s

où s désigne le port à utiliser. Ce qui se produit dépend de la valeur de s :

valeur de s	résulat
moins de 0	SYNTAX ERROR
0	retour à l'unité normale (clavier pour IN#, écran pour PR#)
de 1 à 7	transfert l'entrée ou la sortie de l'unité dont l'interface est insérée dans le numéro de port considéré (appelle le DOS, si l'unité connectée est un DISK II).
de 8 à 16	SYNTAX ERROR, en exécution différée (à partir d'un programme). Le système se "plante" en commande directe.
de 17 à 65535	RANGE ERROR
> 65535	SYNTAX ERROR

La commande IN#0 rétablit les entrées postérieures au clavier; PR#0 rétablit les sorties postérieures à l'écran.

Lorsque le DOS est en fonctionnement, les commandes IN# et PR# peuvent être utilisées en commande directe de la manière habituelle (voir les manuels de BASIC). Mais lorsque ces commandes sont utilisées à l'intérieur d'un programme, elles doivent prendre la forme des commandes du DOS, par exemple :

```
10  D$ = "" : REM CTRL-D
20  PRINT D$; "PR# 1"
30  PRINT D$; "IN# 2"
```

Lorsque le DOS n'est pas en fonctionnement, les commandes IN# et PR# initialisent les registres d'entrée/sortie du Moniteur.

Lorsque le DOS est en fonctionnement, les registres d'entrée/sortie du Moniteur orientent en permanence vers le DOS, et ce sont les registres d'entrée/sortie du DOS qui sont initialisés pour la nouvelle unité d'entrée/sortie choisie.

Voici ce qui arrive à chaque fois qu'un caractère entre dans le micro-ordinateur ou en sort :

Lorsqu'un caractère est envoyé en sortie, le micro-ordinateur le dirige vers le DOS. Si le caractére doit être renvoyé (parce que n'étant pas une commande valide du DOS), avant de le renvoyer, le DOS

1. transfert ses registres d'entrée/sortie du Moniteur. Puis le caractère est envoyé à l'unité qui est maintenant adressée par les registres d'entrée/sortie du Moniteur.

2. se reconnecte, en plaçant de nouveau ses pointeurs dans les registres d'entrée/sortie du Moniteur.

De la même manière, lorsqu'un caractère doit être entré dans le micro-ordinateur, la demande est orientée vers le DOS par les registres d'entrée/sortie du Moniteur. Avant de renvoyer le caractère vers le micro-ordinateur, le DOS fait les mêmes opérations 1 et 2 vues précédemment.

Lorsque le DOS est en fonctionnement, tous les caractères sont interceptés depuis l'unité d'entrée, et avant que le caractère n'ait pu affecter le BASIC simple, le BASIC étendu ou le Moniteur. C'est pourquoi lorsque les commandes IN# ou PR# sont tapées en commande directe au clavier, elles peuvent être captées par le DOS qui peut ainsi faire l'échange des registres.

Le DOS intercepte tous les caractères en sortie du micro-ordinateur, avant qu'ils n'atteignent l'unité de sortie (mais après que les registres d'entrée/sortie du Moniteur aient été affectés).

C'est pourquoi les commandes IN# et PR#, si elles sont données depuis un programme, et non sous la forme des commandes du DOS (précédées d'un CTRL-D), peuvent déconnecter le DOS par le simple fait de changer les registres d'entrée/sortie du micro-ordinateur. Grâce au fait que le DOS restaure les registres d'entrée/sortie à chaque fois qu'un caractère est envoyé, ou reçu, par son intermédiaire, il se reconnecte généralement automatiquement, à condition de n'avoir pas été simultanément déconnecté en lecture et en écriture.

Si vous exécutez une commande PR# depuis un programme, par exemple, après l'exécution de l'instruction

```
50 PR# 1
```

le DOS sera partiellement déconnecté, et dans l'incapacité d'intercepter d'éventuels caractères envoyés en sortie. Mais le DOS est toujours connecté pour les entrées, et le prochain caractère entré aura pour effet de restaurer les registres d'entrée/sortie, donc de reconnecter le DOS pour les sorties.

Il en est de même lors de l'utilisation de la commande IN# depuis un programme, par exemple après l'exécution de l'instruction

```
60 IN# 1
```

le DOS sera déconnecté pour les entrées. Le DOS est toujours connecté néanmoins pour les sorties, et le prochain caractère envoyé en sortie (même un caractère d'appel ou un retour chariot) aura pour effet de le reconnecter à la fois pour l'entrée et pour la sortie.

Pour éviter de telles situations, et permettre au DOS de fonctionner normalement, les commandes IN# et PR# doivent être
- soit données en commande directe,
- soit, si elles sont données dans un programme, sous la forme d'une commande du DOS, par exemple :

```
10  D$ = "" : REM CTRL-D
20  PRINT D$; "PR# 1"
30  PRINT D$; "IN# 2"
```

C'est le caractère CTRL-D qui caractérise les commandes DOS.

TABLE 1 - LES REGISTRES D'ENTREE/SORTIE DU MONITEUR

Registre d'entrée du Moniteur : adresse 56-57 ($38-$39)

Lorsque le contenu du registre est donné par	*A la valeur*	*Les entrées postérieures viennent*
RESET 0 CTRL-K (note 1) IN#0 (note 2)	-741 ($FD1B)	de la routine d'entrée clavier du Moniteur
s CTRL-K (note 1) IN# s (note 2) (étant donné, s>0)	49152 + s * 256 ($Cs00)	du port # s si le port # s contient un interface disquette le DOS est appelé
appel du DOS	-8574 + haut de la mem. (-$217E + $ haut de la mémoire)	du DOS

REGISTRE DE SORTIE DU MONITEUR - ADRESSE 54-55 ($36-$37)

Lorsque le contenu du registre est donné par	*A la valeur*	*Les sorties postérieures vont vers*
RESET 0 CTRL-P (note 1) PR#0 (note 2)	-528 ($FDF0)	La routine de sortie écran du Moniteur
s CTRL-P (note 1) PR# s (note 2) (étant donné, s>0)	49152 + s * 256 ($Cs00)	le port # s si le port # s contient un interface disquette le DOS est appelé
appel du DOS	-8514 + haut de la mem. (-$2142 + $ haut de la mémoire	le DOS

Note 1. Les commandes s CTRL-K et s CTRL-P sont des commandes du Moniteur. Pour taper CTRL-K (ce caractère n'apparaît pas à l'écran), taper la lettre K pendant que la touche CTRL est maintenue enfoncée.

Note 2. Cette commande affectera le contenu du registre du Moniteur uniquement dans le cas où il s'agit d'une instruction dans un programme, et que cette instruction n'est pas donnée sous la forme d'une commande du DOS (précédée d'un PRINT CTRL-D).

Note 3. En plus des commandes citées en Table 1, on peut évidemment modifier les registres du Moniteur, en y mettant, par l'instruction POKE, les valeurs adéqutes pour adresser une unité d'entrée/sortie, ou pour reconnecter le DOS.

TABLE 2 - LES REGISTRES D'ENTREE/SORTIE DU DOS

LE REGISTRE D'ENTREE DU DOS

Lorsque le contenu du registre est donné par	*A la valeur*	*Les entrées postérieures viennent*
l'appel du DOS RESET 3D0G IN# 0 (note 4) PRINT $; "IN# 0" (note 5)	-741 ($FD1B)	de la routine d'entrée clavier du Moniteur
IN# s (note 4) PRINT D$; "IN#s" (note 5) (où s> 0)	49152 + s * 256 ($Cs00)	du port # s si le port # s contient un interface disquette le DOS est appelé

LE REGISTRE DE SORTIE DU DOS

Lorsque le contenu du registre est donné par	*A la valeur*	*Les sorties postérieures vont vers*
l'appel du DOS RESET 3D0G PR# 0 (note 4) PRINTD$;"PR# 0" (note 5)	-528 ($FDF0)	de la routine de sortie écran du Moniteur
PR# s (note 4) PRINTD$; "PR#s" (note 5) (où s> 0)	49152 + s * 256 ($Cs00)	le port # s si le port # s contient un interface disquette le DOS est appelé

Note 4. Cette commande n'affecte pas le contenu des registres d'entrée/sortie du DOS si elle fait partie d'un programme, et si elle n'est pas donnée sous la forme d'une commande du DOS (précédée d'un CTRL-D). Si une ligne de programme telle que :

```
120 PR# 3
```

est exécutée, c'est le contenu du *registre de sortie du Moniteur* qui sera modifié, déconnectant partiellement le DOS jusqu'à la prochaine instruction INPUT.

Note 5. On suppose ici que la variable alphanumérique D$ contient le caractère CTRL-D.

Note 6. Quelle que soit l'unité d'entrée/sortie adressée par les registres d'entrée/sortie du DOS, les entrées peuvent être reçues depuis la disquette, et les sorties envoyées sur la disquette.

Note 7. En plus de commandes de la Table 2, il est possible de modifier les registres d'entrée/sortie du DOS, directement par l'instruction POKE, pour adresser des unités d'entrée/sortie. Il faut cependant tenir compte de deux facteurs :

– l'adresse des registres d'entrée/sortie du DOS varie suivant la taille mémoire du micro-ordinateur;
– l'adresse des registres d'entrée/sortie du DOS varie suivant les versions du DOS.

C'est pourquoi il convient d'utiliser une procédure spéciale pour modifier le contenu des registres d'entrée/sortie du DOS :

a) changer le contenu des registres d'entrée/sortie du Moniteur en y mettant les valeurs que l'on désire mettre dans les registres d'entrée/sortie du DOS (ceci peut être fait par l'instruction POKE, ou par des commandes -non DOS- PR# ou IN# dans un programme).

b) CALL 1002 (à partir du Moniteur, $3EAG)
Après cet appel au sous-programme en langage machine situé en 1002, le contenu des registres d'entrée/sortie du Moniteur sera transféré dans les registres d'entrée/sortie du DOS. L'appel à ce sous-programme peut aussi être utilisé pour reconnecter le DOS dans un programme où il a été nécessaire, pour une raison quelconque, de le déconnecter temporairement (voir en Annexe F, un exemple d'utilisation de ce sous-programme).

Note 8. Les commandes du Moniteur s CTRL-K ou s CTRL-P, lorsqu'elles sont tapées au clavier, ne sont pas reconnues par le DOS : elles affectent directement les registres d'entrée/sortie du Moniteur.

La commande CHAIN en BASIC simple

Certaines applications nécessitent l'exécution de deux ou de plusieurs programmes successivement. Très souvent, le second programme doit utiliser les variables et les tables constituées par le programme. Or, la commande RUN met à zéro toutes les variables et toutes les tables du programme. La commande CHAIN du DOS permet d'exécuter un programme, sans effacer les variables du programme précédent.

Supposons que nous ayons un programme en BASIC simple, et nommé PARTIE 1. Après son exécution, la commande CHAIN PARTIE 2 chargera en mémoire un programme écrit en BASIC simple et nommé PARTIE 2, sans remettre à zéro les variables et les tables du programme PARTIE 1, et l'exécutera. La commande CHAIN peut être utilisée en commande directe, comme montré ci-dessus, ou bien depuis un programme, par exemple la dernière ligne de programme PARTIE 1 pourrait être :

```
20010  D$ = "" : REM CTRL-D
20020  PRINT D$; "CHAIN PARTIE 2"
```

où D$ contient l'habituel CTRL-D.

La syntaxe pour cette commande est :

```
CHAIN f (,Ss) (,Dd) (,Vv)
```

La commande CHAIN et le BASIC étendu

La commande CHAIN ne fonctionne qu'avec les programmes en BASIC simple. Si vous n'avez pas besoin de conserver les variables et les tables d'un programme à l'autre, il est facile d'enchaîner l'exécution des programmes écrits en BASIC étendu. Il suffit que la dernière ligne du premier programme soit :

```
200000 PRINT CHR$ (4);  "RUN PARTIE 2"
```

Lorsque cette ligne est exécutée, le programme PARTIE 2 est chargé (effaçant ainsi le programme PARTIE 1, et toutes ses variables), puis exécuté.

Une procédure différente peut être adoptée pour charger et exécuter successivement plusieurs programmes écrits en BASICS étendu, sans perdre le contenu de leurs variables et de leurs tables.

Pour enchaîner les programmes écrits en BASIC étendu, il faut utiliser un sous-programme, écrit en langage machine, nommé CHAIN, qui est sur la disquette maître du système.

Pour réaliser le chaînage de programmes nommés, par exemple, PARTIE 1 et PARTIE 2, il faut que le programme CHAIN soit sur la même disquette que le programme PARTIE 2.

Les deux dernières lignes du programme PARTIE 1 doivent être :

```
20000 PRINT CHR$ (4);  "BLOAD CHAIN,A520"
20010 CALL  520 "PARTIE 2"
```

Les numéros de lignes peuvent évidemment être différents, mais ces deux instructions sont obligatoirement consécutives, sans aucune autre instruction entre elles deux. La première ligne charge en mémoire le sous-programme de chaînage; la seconde réalise le chaînage.

Attention : il ne doit y avoir aucun espace entre l'adresse du CALL 520 et le guillemet qui le suit. L'adresse du CALL ne doit pas être donnée en notation hexadécimale.

Si vous disposez de l'APPLESOFT en ROM, vous pouvez copier le sous-programme CHAIN sur une disquette en utilisant le programme FID (annexe J).

Si vous utilisez le BASIC étendu APPLESOFT en mémoire (appelé à partir de la disquette), vous pouvez copier le sous-programme CHAIN sur une autre disquette en utilisant le programme FID (annexe J), ou bien procédez de la manière suivante :

à partir de la disquette Maître du système, chargez-le en mémoire par la commande :

```
BLOAD CHAIN,A12296
```

puis sauvegardez-le sur une disquette par la commande :

```
BSAVE CHAIN,A12296,L456
```

Remarquez que les adresses utilisées pour copier le sous-programme CHAIN sont différentes de l'adresse utilisée lors de l'exécution.

ANNEXE A

Les types de fichier et leurs commandes du DOS

104 Les types de fichiers/programmes et les commandes correspondantes
104 Liste des types de fichiers/programmes et des commandes du DOS qui s'y rapportent

Les types de fichiers/programmes et les commandes correspondantes

Dans la plupart des cas, les commandes du DOS peuvent être utilisées soit en commande directe, soit en mode différé. Certaines commandes, cependant, (par exemple READ et WRITE) ne peuvent être utilisées qu'en mode différé.

La plupart des commandes du DOS se réfèrent à un fichier ou à un programme nommément désigné. Le tableau ci-dessous donne les commandes du DOS, classées en ordre alphabétique, en fonction des types de fichiers/programmes auxquelles elles se rapportent.

TABLEAU DES COMMANDES DU DOS, PAR TYPE DE FICHIER AUXQUELLES ELLES SE RAPPORTENT

Commandes du DOS	IProgramme Basic simple	Programme Basic étendu	Fichier mode texte accès séq.	Fichier mode texte accès directl	Fichier mode binaire
APPEND			x		
BLOAD					x
BRUN					x
BSAVE					x
CHAIN	x				
CLOSE			x	x	
DELETE	x	x	x	x	x
EXEC			x		
INIT	x	x			
LOAD	x	x			
LOCK	x	x	x	x	x
OPEN			x	x	
POSITION			x		
READ			x	x	
RENAME	x	x	x	x	x
RUN	x	x			
SAVE	x	x			
UNLOCK	x	x	x	x	x
VERIFY	x	x	x	x	x
WRITE			x	x	

Remarque : les commandes APPEND, OPEN, POSITION, READ, WRITE ne peuvent être utilisées qu'en mode différé.

Liste des types de fichiers/programmes et des commandes qui s'y rapportent

PROGRAMMES EN BASIC SIMPLE EXCLUSIVEMENT :
CHAIN

PROGRAMMES EN BASIC SIMPLE ET EN BASIC ÉTENDU :
INIT
LOAD
SAVE
RUN

FICHIERS EN MODE TEXTE A ACCÈS SÉQUENTIEL EXCLUSIVEMENT :
APPEND
EXEC
POSITION

FICHIERS EN MODE TEXTE, À ACCÈS SÉQUENTIEL OU À ACCÈS DIRECT :
OPEN
CLOSE
READ
WRITE

FICHIERS/PROGRAMMES EN IMAGE MÉMOIRE (BINAIRE) :
BLOAD
BRUN
BSAVE

TOUS TYPES DE FICHIERS :
DELETE
LOCK
UNLOCK
RENAME
VERIFY

Remarque : les commandes APPEND, OPEN, POSITION, READ, WRITE ne peuvent être utilisées qu'en mode différé.

ANNEXE B
Les messages du DOS

108 Les codes d'erreur
109 Analyse des codes d'erreur

Lorsqu'une erreur est détectée par le DOS, un message descriptif est affiché, et l'exécution du programme en cours est interrompue. Les messages du DOS s'ajoutent aux messages du BASIC.

En BASIC étendu, les messages d'erreur sont préccédés d'un point d'interrogation. Exemple :

```
?SYNTAX ERROR
```

En BASIC simple, les messages d'erreur sont précédés de trois astérisques. Exemple :

```
***SYNTAX ERROR
```

Les messages du DOS peuvent se repérer par le fait qu'ils ne sont précédés d'aucun caractère. Exemple :

```
SYNTAX ERROR
```

Un message d'erreur du DOS s'affiche exactement de la même manière, que le programme soit écrit en BASIC simple ou en BASIC étendu ou encore en langage machine.

Si le DOS détecte une erreur lors de l'utilisation d'un programme en langage machine, le mode de fonctionnement revient au type de BASIC (simple ou étendu) qui était utilisé avant l'appel du mode Moniteur, avant l'affichage du message d'erreur du DOS.

Il est possible, grâce à l'utilisation de la commande ONERR GOTO du BASIC étendu (voir le manuel de BASIC étendu), de créer des routines pour maîtriser les cas d'erreur du DOS, et éviter l'interruption du programme.

En effet, si une erreur est détectée par le DOS dans un programme, postérieurement à une commande ONERR GOTO, un code erreur est mémorisé dans l'octet 222 (décimal). Il suffit alors de saisir le code erreur par une instruction PEEK. Par exemple :

```
Y = PEEK(222)
```

met dans la variable Y la valeur du code erreur.

On trouvera ci-dessous les messages du DOS et les codes erreur correspondants, ainsi que la cause la plus probable de la condition d'erreur.

Les codes d'erreur

CODE	MESSAGE DU DOS	CAUSE PROBABLE
1	LANGUAGE NOT AVAILABLE langage non disponible	Le BASIC étendu n'est pas sur disquette.
2, 3	RANGE ERROR hors limite	Paramètre d'une commande hors limite.
4	WRITE PROTECTED protection écriture	L'étiquette de protection d'écriture est en place sur la disquette.
5	END OF DATA fin d'informations significatives	Une lecture a été tentée dans un fichier en mode texte à un endroit où aucun enregistrement n'avait été préalablement écrit.

6	FILE NOT FOUND fichier non trouvé	Mauvaise orthographe du nom du fichier, ou fichier inexistant sur la disquette.
7	VOLUME MISMATCH volume incompatible	Erreur du paramètre
8	I/O ERROR erreur d'entrée/sortie	Porte de la disquette ouverte ou disquette non initialisée.
9	DISK FULL disquette pleine	Il y a trop de fichiers sur la disquette.
10	FILE LOCKED fichier verrouillé	Tentative d'écriture dans un fichier verrouillé.
11	SYNTAX ERROR erreur de syntaxe	Erreur de nom de fichier, de paramètre ou de virgule.
12	NO BUFFERS AVAILABLE pas de mémoire tampon disponible	Il y a trop de fichiers en mode texte ouverts simultanément.
13	FILE TYPE MISMATCH type de fichier incompatible	Le type de fichier est incompatible avec la commande.
14	PROGRAM TOO LARGE programme trop grand	La place disponible en mémoire est insuffisante.
15	NOT DIRECT COMMAND	La commande DOS ne peut être donnée que dans un programme.

Analyse des codes d'erreur

LANGUAGE NOT AVAILABLE (Code d'erreur 1)

Ce message est affiché si le DOS ne peut trouver le langage, que ce soit le BASIC simple ou le BASIC étendu, nécessité pour l'exécution d'une commande.

Les commandes FP, INT et RUN nécessitent la recherche d'un langage, et peuvent donc être à l'origine d'une erreur de code 1.

Dans le cas où le BASIC simple est demandé, le DOS le recherche en ROM. Dans le cas où le BASIC étendu est demandé, le DOS le recherche d'abord en ROM (s'il s'agit de la carte APPLESOFT, *indépendamment* de la position de l'interrupteur de la carte). Si le BASIC étendu est indisponible en ROM, le DOS le recherche sur disquette (le numéro de port et de lecteur sont les derniers utilisés). Si le DOS ne trouve pas le BASIC étendu sur la disquette concernée, il ne le recherche pas sur les éventuels autres lecteurs du système.

Généralement, ce message est affiché lorsque le DOS recherche le BASIC étendu, alors que la disquette concernée ne le comporte pasm. Il convient soit de remplacer par une autre qui contient le BASIC étendu, soit d'utiliser le paramètre D (qui peut l'être avec une quelconque commande du DOS), pour sélectionner un autre lecteur. Par exemple :

FP, D2

Si vous pensez que le DOS aurait dû trouver le BASIC simple en ROM alors qu'il ne l'a pas trouvé, vous pouvez essayer les opérations suivantes :

1. Couper l'alimentation, et retirer le couvercle.
2. Repérer la rangée de mémoires ROM (des boîtiers noirs, rectangulaires, relativement grands) située au centre du circuit imprimé. Ces composants sont référencés "ROM F8", "ROM F0", "ROM E8" et "ROM E0".
3. Les enfoncer.
4. Replacer le couvercle, et essayer à nouveau la commande INT.

Si vous pensez que le DOS aurait dû trouver le BASIC étendu en ROM sur carte, alors qu'il ne l'a pas trouvé, essayez les opérations suivantes :

1. Couper l'alimentation électrique, et retirer le couvercle.
2. Retirer la carte APPLESOFT. Repérer la rangée de cinq mémoires ROM. Elles sont référencées D0, D8, E0, E8 et F0.
3. Les enfoncer.
4. Remettre la carte APPLESOFT dans le port numéro zéro (le port d'extrême gauche).
5. Remettre le couvercle, et essayer à nouveau la commande FP.

RANGE ERROR (code d'erreur 2 ou 3)

Cette erreur se produit lorsque la valeur d'un paramètre d'une commmande du DOS est soit trop petite, soit trop grande. Reportez-vous au manuel pour connaître les paramètres propres à chaque commande du DOS. Vous trouverez ci-dessous les valeurs limites de chacun des paramètres :

Nature du fichier	Paramètre	Symbole	Minimum	Maximum
Tous fichiers	Port	S	1	7
	Lecteur	D	1	2
	Volume	V	0*	254
Fichiers séquentiels en mode texte	Octet	B	0	32767
	Rubrique relative	R	0	32767
	Rubrique absolue *EXEC*	R	0	32767
Fichier à accès direct en mode texte	Longueur d'enregistr.	L	1	32767
	Numéro d'enregistrem.	R	0	32767
Fichiers en mode binaire	Adresse de début	A	0	65535
	Nombre d'octets	L	1	32767

Commande du DOSNombre	Minimum	Maximum	
PR# s	s	0	16**
IN# s	s	0	16**
MAXFILES n	n	1	16

(*) Le plus petit numéro de volume que la commande INIT peut affecter à une disquette est 1.
(**) Le plus grand numéro de port est 7. En exécution en mode différé uniquement (c'est-à-dire à partir d'un programme), le message SYNTAX ERROR est affiché pour une valeur de s supérieure à 7.

Remarque : l'utilisation de valeurs hors des limites pour un paramètre ne donne pas systématiquement l'affichage de RANGE ERROR. Par contre, l'utilisation de toute valeur inférieure à 0 ou supérieure à 65535 donnera systématiquement l'affichage de SYNTAX ERROR (et non de RANGE ERROR).

WRITE PROTECTED (code d'erreur 4)

Ce message est engendré lorsque le DOS tente une écriture sur disquette, et que celle-ci n'a pas l'encoche de protection d'écriture sur le côté droit. Les causes les plus probables sont énumérées ci-dessous :

1. Il y a une étiquette qui recouvre l'encoche de protection d'écriture; c'est une sécurité pour empêcher une écriture involontaire sur la disquette. L'étiquette peut être retirée si l'utilisateur désire lever la protection. La commande SAVE, BSAVE ou WRITE peut alors s'exécuter.

2. La disquette ne comporte pas d'encoche. C'est le cas d'une disquette Maître; elle est ainsi complètement protégée. Il est possible de découper soi-même une encoche, bien que cela ne soit pas recommandé. Il convient de le faire très exactement au bon emplacement. Utilisez une autre disquette comme modèle.

3. Si le message s'affiche lors de l'exécution de programme COPY, et que la cause n'est pas l'un des deux cas que nous venons de voir, il est possible que la disquette ait été mal insérée dans le lecteur (dans tous les autres cas, le DOS affiche le message I/O ERROR à la suite d'une insertion incorrecte de la disquette).

END OF DATA (code d'erreur 5)

Cette condition d'erreur se produit lors d'une tentative de lecture d'un fichier en mode texte à un endroit où aucune information n'a été préalablement écrite. Les octets situés au-delà du dernier enregistrement d'un fichier séquentiel en mode texte, ou au-delà de la dernière rubrique d'un enregistrement à accès direct en mode texte peuvent contenir la valeur zéro. Le zéro est le code ASC II pour un caractère inexistant, ou une absence de caractère. Toute tentative de lecture d'un caractère inexistant a pour effet l'affichage du message END OF DATA. Il convient de se rappeler que seule la commande OPEN ramène automatiquement le pointeur de fichier au premier enregistrement.
L'affichage de ce message d'erreur se produit généralement après un ordre INPUT ou un ordre GET. Il est possible que la cause soit l'une de celles énumérées ci-dessous :

1. Trop d'ordres INPUT successifs ont été exécutés (ou un ordre INPUT suivi de trop de variables). Chaque ordre INPUT exécuté a pour effet la lecture de la prochaine rubrique.

2. Trop d'ordres GET successifs ont été exécutés. Chaque ordre GET exécuté a pour effet la lecture du prochain octet (ou caractère).

3. La valeur du paramètre B (octet) est trop grande. Dans les fichiers à accès séquentiel, ce paramètre ne doit pas désigner un octet au-delà du dernier retour-chariot du fichier. Dans les fichiers à accès direct, ce paramètre ne doit pas désigner un octet situé après le dernier retour-chariot de l'enregistrement concerné. Rappelez-vous que le numéro d'ordre du premier octet à l'intérieur d'un enregistrement est Ø et non 1.

4. La valeur du paramètre R (rubrique relative) d'une commande POSITION est trop grande. Dans les fichiers à accès séquentiel, ce paramètre ne doit pas désigner une rubrique située au-delà de la dernière rubrique du fichier. Dans les fichiers à accès direct, le paramètre R de la commande POSITION ne doit pas désigner une rubrique située au-delà de la dernière rubrique de l'enregistrement considéré.
Il convient de se rappeler que le paramètre R (rubrique relative) de la commande POSITION est distinct du paramètre R de la commande READ. Il désigne une rubrique dans le fichier, *par rapport* à la position du pointeur, et *en avant* de celle-ci exclusivement. RØ signifie que le pointeur doit rester à la même position, R1 que le pointeur doit avancer d'une rubrique (il se positionne au premier octet de la prochaine rubrique), et ainsi de suite.
La commande POSITION déclenche une exploration de la suite du fichier, pour la recherche du Rième retour-chariot. Si un octet nul (caractère inexistant) est rencontré lors de l'exploration, le message END OF DATA est engendré. Dans ce cas aucun ordre INPUT ou GET n'a encore été donné.

5. La valeur du paramètre R (rubrique absolue) dans une commande EXEC est trop grande. Ce paramètre peut désigner la première rubrique au-delà de la dernière rubrique existante, mais pas la seconde et les suivantes. Rappelez-vous que RØ désigne la première rubrique d'un fichier.

6. Le paramètre R dans une commande READ concernant un fichier à accès direct désigne un enregistrement dans lequel aucune information n'a encore été écrite. Il est en effet indispensable que des informations soient écrites au préalable dans un fichier à accès direct, avant de pouvoir être lues.
Rappelez-vous que le paramètre R d'une commande READ est distinct du paramètre R d'une commande POSITION ou EXEC. Le paramètre R de la commande READ désigne un numéro d'enregistrement : RØ est le premier enregistrement du fichier, et ainsi de suite.
Le DOS utilise le paramètre L de la commande OPEN pour calculer l'adresse de début du Rième enregistrement du fichier. C'est pourquoi il est indispensable que la valeur du paramètre L donnée dans la commande OPEN précédant la lecture soit strictement identique à celle qui avait été donnée dans la commande OPEN qui a précédé l'écriture.

FILE NOT FOUND (code d'erreur 6)

Cette erreur est détectée lorsqu'une commmande DOS comporte un fichier dont le nom n'est pas sur la disquette concernée. Seules les commandes SAVE, BSAVE, INIT et OPEN peuvent créer un fichier et insérer un nouveau nom dans le catalogue. Le DOS ne signale pas d'erreur si le fichier spécifié dans la commande CLOSE n'est pas dans le catalogue.
A ces cinq exceptions près, un nom de fichier donné dans une commande du DOS doit obligatoirement correspondre à un nom mémorisé dans le catalogue de la disquette.
Plusieurs situations peuvent générer cette erreur :

1. Il est possible que le nom du fichier soit mal orthographié, à la suite d'une faute de frappe ou parce que la virgule qui sépare le nom de fichier du paramètre qui le suit a été omise. Une vérification dans le catalogue permet de s'en assurer. *ATTENTION :* si vous avez, par erreur, introduit des caractères de contrôle dans le nom du fichier lors de sa création, ceux-ci n'apparaissent pas à l'écran lors de l'affichage du catalogue. Voir l'annexe F à ce sujet.

2. Le fichier est sur une autre disquette.

3. Le fichier a été supprimé par erreur de la disquette.

4. Lors de l'initialisation par INIT, vous indiquez le nom du programme d'appel. Le DOS tente d'exécuter ce programme d'appel chaque fois qu'il est appelé à partir de cette disquette. A moins d'avoir sauvegardé un programme comportant le nom du programme d'appel, le message FILE NOT FOUND sera affiché à chaque fois que le DOS sera appelé. Si vous ne vous souveniez plus du nom du programme d'appel, il convient soit de re-initialiser la disquette, soit de passer le programme MASTER CREATE pour donner un nouveau nom au programme de salutations, sans perdre le contenu de la disquette.

VOLUME MISMATCH (code d'erreur 7)

Cette condition d'erreur fait suite à l'utilisation d'un numéro de volume pour le paramètre V qui est différent de celui de la disquette concernée. Le numéro de volume d'une disquette est affiché en tête du catalogue. Si aucun numéro de volume n'est spécifié dans une commande du DOS, celui-ci est ignoré. Si le numéro spécifié dans une commande du DOS est Ø, celui-ci est aussi ignoré. Si aucun numéro de volume n'est spécifié lors de l'initialisation d'une disquette (INIT), ou si le numéro de volume Ø est spécifié, la disquette initialisée comportera le numéro de volume par défaut 254.

I/O ERROR (code d'erreur 8)

Ce message est engendré après un échec pour lire ou pour écrire sur une disquette; le DOS tente de lire ou d'écrire 96 fois avant de déclarer l'impossibilité.
Différentes situations peuvent déclencher ce message. Elles sont énumérées ci-dessous :

1. La porte du lecteur est ouverte.

2. Il n'y a pas de disquette dans le lecteur sélecté.

3. La disquette n'a pas été initialisée.

4. La disquette est mal insérée dans le secteur.

5. Lors de l'exécution d'une commande VERIFY, le DOS a détecté que le fichier spécifié était incorrectement mémorisé sur la disquette. Si l'information est toujours en mémoire centrale, essayez de l'écrire à nouveau, éventuellement sur une autre disquette.

6. Le paramètre D d'une commande du DOS désigne un lecteur qui n'existe pas dans le système. Le lecteur par défaut est maintenant un lecteur inexistant. Il suffit de spécifier un paramètre D valide dans une nouvelle commande DOS pour rétablir la situation.

7. Le système essaie d'accéder à une disquette formatée à 13 secteurs, alors que le DOS en attend 16. Utilisez le programme MUFFIN pour passer la disquette à 16 secteurs.

8. Le paramètre S d'une commande du DOS désigne un port dans lequel aucun contrôleur de disquette n'est inséré.

Cette situation est gênante. Le numéro de port par défaut est maintenant un port vide. La prochaine commande du DOS donnera le même message d'erreur. Pour redonner le bon numéro de port :
a) frapper la touche RESET;
b) si le caractère (*) du Moniteur apparaît, taper 3DØG;
c) taper CATALOG Ss, s étant le numéro de port correct.

DISK FULL (code d'erreur 9)

Ce message est affiché lorsque le DOS tente de mémoriser des informations sur disquette, et constate qu'il n'y a plus suffisamment de place pour le faire.
Au total, 496 secteurs peuvent être utilisés pour les programmes et les fichiers. On peut voir le nombre de secteurs utilisés lors de l'affichage du catalogue. Si la taille d'un fichier excède 255 secteurs, le DOS reprend le compte à ØØØ.
Si le message DISK FULL est affiché, rassurez-vous : tous les fichiers de la disquette ont été fermés, et le DOS a sauvegardé ce qu'il a pu. S'il s'agissait de la sauvegarde d'un programme, la première chose à faire est de le supprimer de la disquette pleine, et de le sauvegarder sur une autre disquette.
Si, alors que le message DISK FULL vient d'être affiché, vous essayez immédiatement de donner une commande SAVE, BSAVE ou WRITE, avant d'avoir supprimé d'autres fichiers, une conséquence curieuse en résultera : le nombre de secteurs de l'une des entrées du catalogue sera remis à zéro. Bien que gênant, ceci n'est pas dangereux : les informations de l'entrée concernée ne sont aucunement affectées. Pour éviter de se trouver dans de telles situations, supprimez des programmes ou des fichiers dès que vous tombez en "DISK FULL".

FILE LOCKED (code d'erreur 10)

Cette condition d'erreur intervient lorsque l'on tente une écriture (SAVE, BSAVE, WRITE, DELETE) d'un programme ou d'un fichier dont le nom a été préalablement verrouillé par un ordre LOCK.
Généralement un nom de fichier ou de programme a été verrouillé pour éviter une modification ou une suppression involontaire. Il convient soit d'utiliser une autre disquette, soit de déverrouiller le fichier.

SYNTAX ERROR (code d'erreur 11)

Une erreur de grammaire a été détectée dans une commande du DOS. Il convient, dans un premier temps, de vérifier la syntaxe dans le manuel. En particulier, il peut s'agir d'un nom de fichier invalide (voir annexe F), d'un paramètre invalide, d'un séparateur (généralement la virgule) incorrect ou absent. Ce message sera affiché si la valeur d'un paramètre est négative ou supérieure à 65535.

Il est invraisemblable que chaque commande du DOS donne lieu à la détection d'une erreur de syntaxe pour l'un des motifs ci-dessus. Généralement, une détection systématique d'erreurs de syntaxe signifie que le DOS est, pour un motif quelconque, déconnecté. Vous pouvez essayer de le récupérer en frappant la touche RESET, puis en tapant 3D0G. Si cette tentative n'est pas couronnée de succès, il ne reste plus qu'à appeler à nouveau le DOS (en perdant le programme éventuellement en mémoire...).

NO BUFFERS AVAILABLE (code d'erreur 12)

Cette condition d'erreur est atteinte lorsqu'une commande du DOS nécessite pour son accomplissement l'utilisation d'une mémoire tampon et qu'aucune mémoire tampon n'est disponible.

Rappelons que lors de son appel, le DOS prévoit trois mémoires tampon. Une commande MAXFILES postérieure peut diminuer ou augmenter le nombre de mémoires tampon prévues. Une mémoire tampon utilisée pour un fichier redevient disponible pour un autre fichier après l'exécution de l'ordre CLOSE correspondant.

Plusieurs commandes du DOS nécessitent l'utilisation d'une mémoire tampon pendant leur exécution. La mémoire tampon est de nouveau disponible à la fin de l'exécution de l'ordre.

Lorsqu'un fichier en mode texte est ouvert par une commande OPEN, une mémoire tampon lui est allouée. Cette mémoire lui reste affectée jusqu'à ce qu'un ordre CLOSE (soit concernant un fichier spécifique, soit un ordre CLOSE général concernant tous les fichiers), intervienne, et libère les mémoires tampon préalablement allouées.

Il convient de se rappeler qu'un fichier en mode texte n'est pas automatiquement fermé par le simple fait que le programme qui l'exploitait soit terminé.

Pour utiliser au mieux les mémoires tampon, on peut fermer les fichiers dès qu'il n'est plus indispensable qu'ils soient ouverts. Il faut se rappeler cependant que l'ordre OPEN repositionne le pointeur de fichier en début de fichier.

Remarquez que la commande MAXFILES peut être utilisée pour étendre le nombre de mémoires tampon avant l'écriture ou le chargement de programme en mémoire. Augmenter le nombre de MAXFILES a pour résultat de baisser l'adresse du HIMEM (adresse haute de la mémoire), donc de détruire totalement ou partiellement un programme écrit en BASIC simple, ou les chaînes de caractères utilisées par un programme écrit en BASIC étendu. C'est la raison pour laquelle il est particulièrement recommandé de ne pas utiliser la commande MAXFILES dans un programme.

FILE TYPE MISMATCH (code d'erreur 13)

Ce message d'erreur est affiché lorsque le DOS tente d'utiliser un nom de fichier qui est déjà assigné, et dont le mode est inadapté à la commande DOS concernée.

Si, après vérification, la commande DOS s'avère correcte, vous pouvez :
- utilisez un nom de fichier qui n'existe pas déjà sur la disquette,
- changer de disquette,
  donner un nouveau nom de fichier à l'aide de la commande RENAME,
- supprimer le fichier par une commande DELETE.

En fait, l'origine de l'erreur est sans doute une combinaison invalide des commandes DOS avec les fichiers existants. On trouvera ci-dessous le tableau des combinaisons correctes :

LOAD f, RUN f, SAVE f	f doit être un programme BASIC simple ou BASIC étendu.
CHAIN f	f doit être un programme BASIC simple
OPEN f, READ f, WRITE f, APPEND f, POSITION f, EXEC f	f doit être un fichier en mode texte.
BLOAD f, BRUN f, BSAVE f	f doit être un programme en langage machine, o un fichier en mode binaire.

Le nom d'un programme de Boot utilisé lors de l'INIT doit correspondre à un programme écrit en BASIC simple ou en BASIC étendu.

PROGRAM TOO LARGE (code d'erreur 14)

Cette condition d'erreur intervient lorsqu'une commande du DOS tente de charger en mémoire un programme à partir de la disquette, et constate que la mémoire disponible n'est pas suffisante pour contenir en totalité le programme à charger.

Il est possible que le HIMEM ait été assigné à une adresse trop basse par un programme précédent par exemple, ou bien que le nombre donné pour MAXFILES ait eu le même effet.

Si vous réassignez trois mémoires tampon par la commande MAXFILES 3, la valeur du HIMEM redeviendra celle qui est donnée en Table 2, Annexe D.

ATTENTION : si dans un programme écrit en BASIC simple, la valeur du HIMEM est abaissée, par exemple pour protéger l'image en mémoire d'une page de graphique haute résolution, il est possible que vous ayez des difficultés lorsque vous passez en BASIC étendu disquette.

Le BASIC étendu en provenance de la disquette occupe environ 12,5 k de mémoire. Le fait de passer en BASIC étendu (par FP, LOAD ou RUN) *ne modifie pas* l'adresse de HIMEM. Lorsque le DOS tente le chargement du BASIC étendu à partir de la disquette, le message "PROGRAM TOO LARGE" sera affiché si l'adresse préalablement fixée pour HIMEM est inférieure à 13100 environ. Après l'affichage du message, le système est toujours en BASIC simple. Il convient donc de restaurer la valeur du HIMEM à partir du BASIC simple. Voir l'Annexe D, table 2, pour la détermination du HIMEM maximum lors de l'utilisation avec le DOS et trois mémoires tampon.

Pour déterminer si un programme tient en mémoire, le DOS considère le nombre entier de secteurs occupés. En général, le dernier secteur n'est pas totalement occupé, mais le DOS n'en tient pas compte pour cette évaluation. Le DOS compare seulement l'octet de fort poids (base 256) de l'adresse LOMEM (BASIC simple) ou de l'adresse HIMEM (BASIC étendu) avec l'octet de fort poids de l'adresse de fin de programme à charger. C'est la raison pour laquelle un programme qui tient en mémoire, mais pour lequel il reste moins de 256 octets de mémoire disponible après chargement, peut donner l'affichage "PROGRAM TOO LARGE". Il est parfois possible de contourner cette difficulté en modifiant légèrement l'adresse de HIMEM ou de LOMEM, avant le chargement du programme.

NOT DIRECT COMMAND (code d'erreur 15)

Cette erreur est détectée l'utilisateur tente une commande DOS concernant un fichier en mode texte en commande directe : APPEND, OPEN, POSITION, READ ou WRITE.

Ces commandes du DOS sont réservées à une utilisation en mode différé, c'est-à-dire uniquement dans un ordre PRINT inséré dans un programme.

ANNEXE C
La mémorisation des informations sur disquette

118 Généralités sur la mémorisation
118 L'écriture d'un fichier à accès séquentiel en mode texte
119 L'écriture d'un fichier à accès direct en mode texte
120 Comment le DOS procède lors de l'écriture des fichiers en mode texte
121 Le contenu des secteurs d'un fichier
121 La liste pistes/secteurs
122 Le catalogue du contenu d'une disquette
125 La table du volume
126 Le schéma d'occupation des pistes
128 Ordre d'affectation des pistes et des secteurs
129 Le chargement d'informations à partir de la disquette
129 La lecture d'un fichier à accès séquentiel
130 La lecture d'un fichier à accès direct

Rappelons qu'un signe $ ou l'abréviation "HEX" indique que le nombre qui suit est en notation hexadécimale.

Généralités sur la mémorisation

Dans un système DISK II, l'information mémorisée sur disquette est enregistrée sur 35 zones concentriques nommées des *pistes*. Les pistes sont numérotées depuis l'extérieur, piste $00, vers l'intérieur, piste $22. Lorsque la disquette est en rotation, la tête de lecture/écriture peut se positionner au-dessus de l'une quelconque des 35 pistes. Chaque piste est divisée en 16 segments, nommés des *secteurs*. Les secteurs sont numérotés de $0 à $F. Chaque secteur peut contenir jusqu'à 256 octets, soit $100 octets d'informations. Lorsque la tête de lecture/écriture est positionnée sur une piste déterminée, les 16 secteurs de la piste passeront sous la tête de lecture à chaque rotation de la disquette. Le DOS écrit toujours sur la disquette par blocs de 256 octets, remplissant à chaque fois complètement un secteur.

Avant de réaliser l'écriture effective sur disquette, le DOS mémorise d'abord les octets à être écrits dans une zone de 256 octets de la mémoire centrale, réservée pour le fichier concerné, et nommée la *mémoire tampon* (en américain, buffer).

Lorsque la mémoire tampon est pleine, un secteur est écrit sur la disquette. Puis le DOS continue à renvoyer les prochains octets à écrire dans la mémoire tampon, jusqu'à concurrence de 256, et ainsi de suite.

Les programmes, ou les fichiers en mode texte, seront écrits par le DOS dans le premier secteur disponible de la disquette. Lorsque le premier secteur d'un fichier est écrit, le second peut être placé par le DOS dans le prochain secteur disponible; celui-ci se trouve peut-être sur une toute autre piste. Les informations constituant un programme ou un fichier ne sont donc pas mémorisées en une suite continue de secteurs.

Au fur et à mesure de l'écriture, le DOS constitue une liste des secteurs et des pistes utilisés pour le fichier considéré. Puis le DOS mémorise cette liste, nommée la *liste pistes/secteurs* (track/sector list) dans un autre secteur (ou plusieurs secteurs) disponible sur la disquette. Le nom du fichier, le type du fichier, sa longueur exprimés en termes de nombre de secteurs, et l'adresse sur disquette de la liste piste/secteurs pour ce fichier sont mémorisés dans un endroit particulier de la disquette situé en piste $11, et nommé le *répertoire* (directory). Simultanément, le *schéma des pistes* (track bit map) exprimé en bits, est mis à jour, afin de représenter fidèlement les secteurs occupés sur chaque piste.

L'écriture d'un fichier à accès séquentiel en mode texte

Les informations enregistrées dans un fichier en mode texte sont en fait une suite de valeurs ASCII (de 1 à 32767 valeurs ASCII) dont la dernière valeur est le caractère « retour chariot », soit ASCII $0D, soit ASCII $8D. Chaque suite ainsi constituée est appelée une *rubrique*.

Dans un fichier séquentiel en mode texte (c'est-à-dire pour lequel un paramètre L, longueur, n'a pas été indiqué dans la commande OPEN), chaque rubrique est enregistrée immédiatement à la suite de la précédente (voir chapitre 6). Le DOS écrit le premier octet d'une rubrique immédiatement après le caractère « retour chariot » concernant la rubrique précédente (à moins que le paramètre B ne soit utilisé). Chaque fois que le fichier est ouvert par une commande OPEN, le DOS assigne le pointeur de fichier à l'octet 0, et recommence donc l'écriture depuis le début du fichier, comme s'il était vide (à moins que le paramère B ne soit utilisé).

Pour écrire à nouveau seulement une rubrique, ou une partie d'une rubrique, dans un fichier à accès séquentiel, la commande WRITE peut être utilisée avec le paramètre B pour commencer l'écriture à l'octet spécifié en valeur absolue, le premier octet du fichier ayant le numéro d'ordre Ø.

L'octet spécifié pour le paramètre B peut être situé avant ou après l'octet sur lequel est positionné le pointeur de fichier.

Il faut remarquer qu'il est assez difficile de se repérer très exactement au niveau d'un octet dans un fichier à accès séquentiel. C'est la raison pour laquelle l'utilisation du paramètre B n'est pas recommandée pour les fichiers à accès séquentiel.

La commande POSITION peut être utilisée avec un paramètre R pour lequel il convient de donner le nombre de rubriques à additionner (et non à soustraire) au pointeur de fichier. Un programme tel que :

```
120  PRINT D$; "OPEN NOMS"
130  PRINT D$; "POSITION NOMS, R13"
140  PRINT D$; "WRITE NOMS"
160  PRINT "APPLE II"
170  PRINT D$; "CLOSE NOMS"
```

aura pour effet l'écriture des caractères APPLE II dans le fichier NOMS à partir du premier octet de la 14e rubrique (puisque la première rubrique a un numéro d'ordre égal à Ø).

Il faut remarquer que la commande POSITION fait avancer le pointeur de fichier du nombre de rubriques spécifiées, et le repositionne sur le premier octet de la nouvelle rubrique. Cependant, si vous voulez écrire à nouveau cette rubrique, après une mise à jour, par exemple, il est essentiel qu'un *nombre exactement égal de caractères* soit écrit lors de la seconde écriture. En effet, si la seconde écriture comporte moins de caractères que la première, ce sont deux rubriques, et non une seule, qui sont créées : celle que l'on vient d'écrire, suivie d'un retour chariot, suivi de la fin de la rubrique d'origine, suivie d'un retour chariot. Si, par contre, la seconde écriture comporte plus de caractères que la première, le retour chariot de la première rubrique est effacé et les premiers caractères de la rubrique suivante le sont aussi. On obtient donc, en une seule rubrique, l'écriture qui vient d'être réalisée et la fin de la rubrique suivante.

L'écriture d'un fichier à accès direct en mode texte

Lorsqu'il s'agit d'un fichier destiné à être traité en accès direct, les *enregistrements* sont nécessairement de longueur fixe. Un enregistrement peut être constitué par une ou plusieurs rubriques. Chaque enregistrement est traité par le DOS comme une entité. La longueur des enregistrements est donnée au DOS par le paramètre L de la commande OPEN. En fait, chaque enregistrement d'un fichier à accès direct est comme un fichier à accès séquentiel dont la longueur totale maximum est celle spécifiée par le paramètre L. Tant que l'on reste à l'intérieur de la limite de longueur maximum, on peut écrire un enregistrement autant de fois que nécessaire, sans affecter aucunement les autres enregistrements du fichier.

La commande WRITE doit être utilisée avec le paramètre R, donnant le numéro d'ordre de l'enregistrement que l'on désire écrire à l'intérieur du fichier (le premier enregistrement a un numéro d'ordre Ø). La commande WRITE peut aussi être utilisée avec le paramètre B, qui spécifie le numéro d'octet à l'intérieur de l'enregistrement à partir duquel l'écriture doit commencer.

Étant donné qu'une *quelconque commande* du DOS termine une écriture (WRITE), il n'est pas possible d'utiliser la commande POSITION pour sauter des rubriques à l'intérieur de l'enregistrement concerné par l'écriture.

Le DOS utilise la longueur spécifiée par le paramètre L pour calculer l'emplacement précis du premier octet de l'enregistrement à écrire. Cet emplacement est calculé par rapport au premier octet de l'enregistrement précédent. Si l'enregistrement précédent a une longueur en nombre d'octets inférieure à L, les octets non significatifs resteront enregistrés à leur place, sans être affectés par l'écriture (voir à ce sujet la section suivante, pour plus de détails).

Si l'on tente d'écrire plus d'octets pour un enregistrement d'un fichier à accès direct qu'il n'en a été spécifié par le paramètre L, tout a l'air de se passer correctement au niveau de l'enregistrement. Cependant, lorsque l'on écrit l'enregistrement suivant, le DOS calcule l'octet de départ de l'écriture à partir du paramètre L. Le nouvel enregistrement efface donc non seulement les caractères supplémentaires de l'enregistrement suivant, mais aussi le retour chariot. Le fichier en accès direct devient inexploitable par le DOS. Une situation à éviter absolument!

Comment le DOS procède lors de l'écriture des fichiers en mode texte

Lors de l'écriture d'une rubrique dans un fichier en mode texte, le DOS vérifie au préalable que le secteur destiné à recevoir les informations a déjà fait l'objet d'une écriture au moins. Si le fichier ne s'est encore jamais étendu sur ce secteur, le DOS écrit des zéros dans les 256 octets de la mémoire tampon de ce fichier, puis il mémorise les informations à écrire dans cette mémoire tampon.

Le contenu de la mémoire tampon est écrit sur la disquette lorsque les 256 octets ont été remplis, ou lors de la fermeture du fichier par la commande CLOSE.

C'est la raison pour laquelle les octets non utilisés d'un secteur dans lequel on vient d'écrire pour la première fois un enregistrement pour le fichier concerné contiennent des zéros (la valeur ASCII de zéro). Une tentative de lecture de ces octets inutilisés a pour conséquence l'affichage du message END OF DATA ERROR, et l'interruption du programme.

Par contre, si le DOS constate que des informations ont déjà été enregistrées dans ce secteur pour le fichier concerné, une lecture du secteur est effectuée, et les informations sont enregistrées dans la mémoire tampon. Après une modification éventuelle de ces informations par l'intermédiaire des commandes DOS WRITE, POSITION (fichier à accès séquentiel exclusivement) et PRINT, le contenu de la mémoire tamp0on est écrit à nouveau sur la disquette, au même emplacement. L'écriture effective à partir de la mémoire tampon est réalisée
- avant l'écriture d'un octet non contenu dans le secteur,
- ou lors d'une commande CLOSE.

Si après avoir écrit une information longue dans un fichier, on la remplace par une information plus courte, et que le DOS écrive cette information dans un secteur qui a déjà été utilisé pour le fichier, les octets inutilisés *ne seront pas réinitialisés avec des zéros.* Les octets qui n'ont pas été remplis à nouveau contiendront les informations qui y avaient été enregistrées lors de l'écriture précédente. Cette remarque s'applique aussi bien dans le cas des octets inutilisés à la fin d'un fichier à accès séquentiel que dans le cas des octets inutilisés à la fin d'un enregistrement dans le cas d'un fichier à accès direct.

Le contenu des secteurs d'un fichier

L'information mémorisée dans chaque secteur dépend de la nature des informations qui y sont enregistrées.

Type d'information	Secteur	Octet (hex)	Contenu
BASIC	1°	0	Longueur du programme : octet de faible poids
		1	Longueur du programme : octet de fort poids
		2 à FF	Programme codé condensé
	secteurs suivants	tous octets	Programme codé condensé
TEXT	tous	tous octets	Représentation des caractères par leur valeur ASCII. Un caractère par octet ($00 indique la fin du fichier)
BINAIRE	1°	0	Adresse de début en RAM : octet de faible poids
		1	Adresse de début en RAM : octet de fort poids
		2	Longueur de la zone en RAM : octet de faible poids
		3	Longueur de la zone en RAM : octet de fort poids
		4 à FF	Information en binaire
	secteurs suivants	tous octets	Information en binaire

La Liste pistes/secteurs

Lorsqu'un fichier est mémorisé sur disquette, le DOS établit une liste des adresses occupées sur la disquette par le fichier. Puis cette liste est elle-même mémorisée sur la disquette, de la même manière que le fichier l'a été.

On trouvera ci-dessous le contenu de la liste pistes/secteurs.

Octet (hex)	Première piste d'une liste pistes/secteurs. Contenu de l'octet
0	Inutilisé.
1	Pointeur de chaînage : numéro de piste contenant la suite de la liste.
2	Pointeur de chaînage : numéro de secteur contenant la suite de la liste (si les deux pointeurs de chaînage sont égaux à zéro, il n'y a pas de chaînage).

3 à 4	Inutilisé.
5-6	Nombre de secteurs de base (nombre de groupes de 122 secteurs.
7 à B	Inutilisé.
C	Numéro de piste du premier secteur du fichiers.
D	Numéro de secteur du premier secteur du fichier.
E	Numéro de piste du second secteur du fichier.
F	Numéro de secteur du second secteur du fichier.
10	Numéro de piste du troisième secteur du fichier.
11	Numéro de secteur du troisième secteur du fichier.
•	• •
•	• •
•	• •
FE	Numéro de piste du 122e secteur du fichier.
FF	Numéro de secteur du 122e secteur du fichier.

Une paire d'octets piste/secteur contenant 0/0 caractérise un secteur non assigné (généralement la fin du fichier, bien que les fichiers en mode texte peuvent contenir des indicateurs 0/0 pour des secteurs non encore assignés où des informations seront ultérieurement écrites).

Si la liste pistes/secteurs concerne plus de 122 paires de pistes/secteurs, les secteurs suivants ont le même format. Les pointeurs de chaînage dans les octets 1 et 2 pointeront éventuellement sur le secteur suivant. S'il s'agit du dernier secteur, les pointeurs de chaînage seront à zéro.

Pour un fichier en mode texte, seules les paires de pistes/secteurs contenant effectivement des informations apparaissent comme différentes de zéro dans la liste pistes/secteurs. Le DOS calcule la *position exacte* de paires pistes/secteurs au sein de la liste, affectant les paires non encore utilisées à zéro.

Prenons un exemple. Supposons que nous ayons un fichier à accès direct dont la longueur des enregistrements (paramètre L) est déclarée être de 128, soit permettant de mémoriser deux enregistrements par secteur, et que l'utilisateur n'ait écrit que le 2700e enregistrement. Treize secteurs du fichier sont effectivement utilisés : l'un contenant le 2700e enregistrement, les doze autres contenant la liste pistes/secteurs. Il est possible qu'ultérieurement le fichier comporte des informations dans les enregistrements 0 à 2683 : elles occuperont alors 1.342 secteurs. Mais tant que ces enregistrements ne sont pas effectivement écrits, aucune place n'est réservée sur la disquette à cet effet. La liste pistes/secteurs comportant les adresses des enregistrements 0 à 2683 occupe 11 secteurs.

Le catalogue du contenu d'une disquette

Lors de l'initialisation d'une disquette, la piste $11 est réservée pour la liste du contenu de la disquette. Le DOS y mémorisera, pour chaque fichier/programme, son nom, son type, le nombre de secteurs occupés (MOD 256), et l'adresse sur disquette de la liste pistes/secteurs du fichier. La commande CATALOG permet de visualiser la majorité de ces informations à l'écran. Chaque secteur du catalogue est formaté comme indiqué ci-dessous :

Octet (hex)	Un secteur du catalogue de la disquette. Contenu des octets
0	Inutilisé.
1	Lien de chaînage : numéro de piste où la suite du catalogue est mémorisé (normalement $11).
2	Lien de chaînage : numéro de secteur où la suite du catalogue est mémorisée (si les deux octets de chaînage sont à zéro, il n'y a pas de chaînage).
de 3 à A	Inutilisé.
de B à 2D	Description du fichier 1 (voir ci-après).
de 2E à 50	Description du fichier 2.
de 51 à 73	Description du fichier 3.
de 74 à 96	Description du fichier 4.
de 97 à B9	Description du fichier 5.
de BA à DC	Description du fichier 6.
de DD à FF	Description du fichier 7.

Les numéros de fichier sont arbitraires. Lorsqu'un fichier est supprimé, le DOS met l'indication de suppression dans la description du fichier concerné (voir ci-après). Lorsqu'un nouveau fichier est mémorisé, le DOS remplace la description du fichier supprimé par la nouvelle desdription à mémoriser. C'est pourquoi, bien qu'à l'origine le catalogue soit constitué en ordre chronologique de créations, l'ordre devient rapidement non significatif.

Le catalogue de la disquette débute en piste $11, secteur $F. En cas de besoin de place supplémentaire, le secteur $F est chaîné au secteur $E. Si cette place n'est plus suffisante, le secteur $E est chaîné au secteur $D. Ce système permet de mémoriser un maximum de 105 fichiers/programmes dans le catalogue. Chaque description de fichier est insérée selon le format ci-après :

Octet pos. relative	Description d'un fichier Contenu des octets
0	Numéro de piste de liste pistes/secteurs (la valeur initiale est mémorisée en 20, et la valeur de l'octet 0 devient $FF lorsque le fichier est supprimé).
1	Numéro de secteur de la liste pistes/secteurs.
2	Type de fichier (voir plus loin en annexe C).
3 à 20	Nom du fichier.
21-22	Nombre de secteurs (MOD 256) occupés par le fichier.

La position relative donne l'emplacement de chaque octet à l'intérieur de chaque description de fichier. Pour trouver l'adresse absolue correspondante, il convient d'additionner l'adresse de début de la description du fichier (table précédente) à la position relative.

C'est parce que le nombre de secteurs occupés par un fichier est mémorisé dans un octet que le nombre maximum est 255 ($FF). Si le nombre de secteurs occupés excède 255, le compteur de secteurs (affiché par le catalogue) repart de 000. Ceci n'a aucune conséquence dans l'utilisation du fichier, mais peut introduire en erreur quant à la place encore disponible sur la disquette.

Examinons maintenant le contenu de l'octet de position relative 2 qui indique le type de fichier.

Octet type de fichier		
Bit	Symbole dans le catalogue	Type de fichier
7	*	Le fichier est verrouillé si ce bit est à 1; il est déverrouillé si ce bit est à 0
6		Disponible pour utilisation ultérieure (normalement à 0)
5		Disponible pour utilisation ultérieure (normalement à 0)
4		Disponible pour utilisation ultérieure (normalement à 0)
3		Disponible pour utilisation ultérieure (normalement à 0)
2	B	Fichier en mode binaire si ce bit est à 1
1	A	Programme en BASIC étendu (APPLESOFT) si ce bit est à 1
0	I	Programme en BASIC simple si ce bit est à 1
	T	Fichier en mode texte si les bits 0 à 6 sont tous à 0

le type de fichier est déterminé par le fait que l'un des bits de l'octet est à 1. Si les bits 0 à 6 sont à zéro, le type par défaut est le mode texte.

Compte tenu du tableau précédent, la valeur de l'octet type de fichier peut donc être :

Valeur de l'octet "type de fichier"		
Type	Fichier déverrouillé	Fichier verrouillé
Texte	0	80
BASIC simple	1	81
BASIC étendu	2	82
Binaire	4	84

La table du volume

Le secteur $Ø de la piste $11 contient la table du volume nommée VTOC (Volume Table Of Contents). Les informations suivantes sont mémorisées dans la VTOC :

VTOC		
Octet (hex)	Valeur (hex)	Description
Ø	2	Non utilisé
1	11	Numéro de la première piste contenant le catalogue
2	ØF	Numéro du 1er secteur contenant le catalogue
3	4	Numéro de version du DOS
4	Ø	Non utilisé
5	Ø	Non utilisé
6	de 1 à FE	Numéro de volume ($FE) par défaut)
de 7 à 26	Ø	Non utilisé
27	7A	Nombre maximum de paires pistes/secteurs pouvant être mémorisées dans chaque secteur de la liste pistes/secteurs
de 28 à 2F	Ø	Non utilisé
3Ø	FF	Ces quatre octets sont un masque pour le schéma d'occupation des pistes/secteurs
31	FF	
32	ØØ	
33	ØØ	
34	23	Nombre de pistes par disquette
35	ØF	Nombre de secteurs par piste
36	ØØ	Nombre d'octets par secteur, octet unitaire
36	ØØ	Nombre d'octets par secteur, octet de faible poids
37	Ø1	INombre d'octets par secteur, octet de fort poids
de 38 à 3B	Ø	occupation de la piste Ø — ces pistes ne sont
de 3C à 3F	Ø	occupation de la piste 1 — pas disponibles
de 4Ø à 43	Ø	occupation de la piste 2 — pour l'utilisateur
44 et 45	?	occupation de la piste 3
46 et 47	Ø	occupation de la piste 3
48 et 49	?	occupation de la piste 4

4A et 4B	0	occupationde la piste 4
/	/	/
/	/	/
/	/	/
78 et 79	?	occupation de la piste $10
7A et 7B	0	occupation de la piste $10
de 7C à 7F	0	occupation de la piste $11 (catalogue et VTOC)
80 et 81	?	occupation de la piste $12
82 et 83	0	occupation de la piste $12
/	/	/
/	/	/
/	/	/
C0 et C1	?	occupation de la piste $22
C2 et C3	0	occupation de la piste $22
de C4 à FF	0	Inutilisé

Le schéma d'occupation des pistes

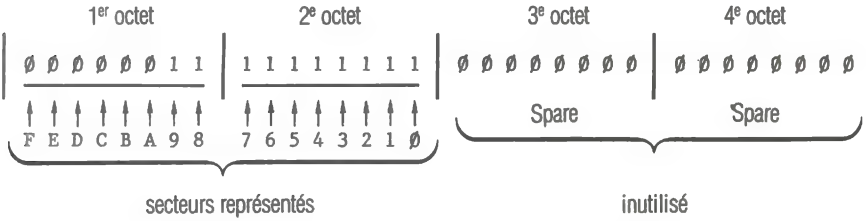
A partir de l'octet $38 de la VTOC (voir table précédente), chaque groupe de quatre octets contient le schéma d'occupation pour chacune des 35 pistes de la disquette. La valeur (0 ou 1) de chaque bit indique au DOS quels sont les secteurs de la piste qui sont libres et quels sont les secteurs qui sont occupés.

C'est le schéma suivant qui est utilisé pour chaque piste :

Schéma d'occupation pour une piste					
Octet (Hex)	Bit	Secteur (Hex)	Octet	Bit	Secteur
1°	7	F	2°	7	7
	6	E		6	6
	5	D		5	5
	4	C		4	4
	3	B		3	3
	2	A		2	2
	1	9		1	1
	0	8		0	0
			3° et 4°	tous	inutilisés

Si un bit du schéma d'occupation d'une piste contient la valeur 1, le secteur correspondant est libre; s'il contient Ø, le secteur correspondant est utilisé. Les bits «non utilisés» dans la table ci-dessus contiennent la valeur Ø.

Un schéma typique d'occupation de piste pourrait apparaître comme suit :



Lorsqu'un fichier est mémorisé sur disquette par l'intermédiaire d'un ordre WRITE, SAVE ou BSAVE, la place est allouée piste entière par piste entière (si possible). Lorsque le fichier est fermé (par un ordre CLOSE), les secteurs non utilisés sont repérés comme tels sur le schéma d'occupation de la piste.

Remarquez que les secteur utilisés par un fichier ne sont libérés qu'à la *suppression du fichier*. Supposons qu'une disquette contienne un programme BASIC nommé GRAND, occupant 1ØØ secteurs. Remplaçons ce programme par un autre programme nommé GRAND, mais occupant seulement 2 secteurs, par un ordre SAVE. Le catalogue montrera que le nouveau programme occupe toujours 1ØØ secteurs.

Pour libérer les secteurs affectés au programme GRAND, mais inutilisés, il faut adopter la séquence suivante des commandes :

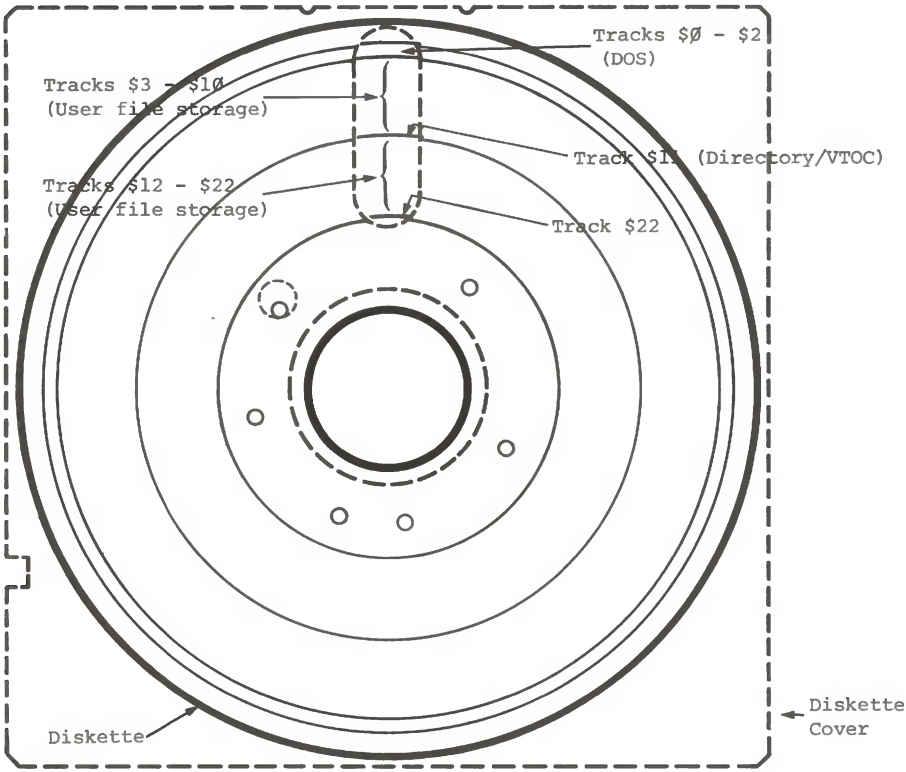
```
LOAD GRAND
DELETE GRAND
SAVE GRAND
```

Un procédé similaire peut être utilisé pour libérer de la place dans les fichiers binaires.

Pour libérer de la place inutilisée dans un fichier en mode texte, deux procédés peuvent être utilisés :
– si le fichier tient en totalité dans une table en mémoire, il faut le lire rubrique par rubrique, et le mémoriser dans la table. Puis supprimer le fichier de la disquette. Enfin réécrire le fichier, à partir de la table en mémoire, sur la disquette;
– si le fichier ne tient pas en mémoire, on peut le lire rubrique par rubrique et, au fur et à mesure, le réécrire sur disquette sous un autre nom de fichier. Lorsque la lecture/écriture est terminée, on peut supprimer l'ancien fichier (DELETE), puis redonner le nom d'origine au second (RENAME).

Ordre d'affectation des pistes et des secteurs

Chaque disquette contient 35 pistes, dont 3 sont réservées au DOS et au catalogue / VTOC. Il reste donc 31 pistes disponibles pour l'utilisateur. Chaque piste contient 16 secteurs. Il y a donc 31 x 16, soit 496 secteurs disponibles pour l'utilisateur. Les secteurs sont affectés en commençant par le secteur $F, remontant jusqu'au secteur $Ø. Les pistes sont affectées en commençant par le secteur $F, remontant jusqu'au secteur $Ø. Les pistes sont affectées en commençant par la piste $12, immédiatement à l'intérieur de la piste du catalogue de la VTOC, et allant vers l'intérieur jusqu'à la piste $22 (la piste la plus intérieure). Lorsque la piste $22 est pleine, l'affectation reprend à la piste $1Ø (immédiatement à l'extérieur de la piste du catalogue de la VTOC), et va jusqu'à la piste $3 (la piste la plus extérieure disponible pour l'utilisateur).



Ordre d'affectation des pistes
d'abord $12 → $22
puis $1Ø → $Ø3

Ordre d'affectation des secteurs
SF → $Ø

Le chargement d'informations à partir de la disquette

Pour charger un fichier à partir d'une disquette, le DOS utilise le même système que pour le mémoriser, mais inversé. Par exemple, après une commande telle que :

LOAD PROGRAMME
ou BLOAD PROGRAMME,

le DOS explore le catalogue situé en piste $11 et trouve l'élément nommé «PROGRAMME». Le catalogue donne alors l'adresse (piste et secteur) de la liste pistes/secteurs de l'élément «PROGRAMME». Le DOS se réfère alors à la liste pistes/secteurs et lit la première paire piste/secteur. Cette paire donne l'adresse du premier secteur contenant l'élément nommé «PROGRAMME». Lorsque le DOS a lu le premier secteur, il se réfère à nouveau à la liste pistes/secteurs pour connaître l'adresse du second secteur de «PROGRAMME», et ainsi de suite.

La lecture d'un fichier à accès séquentiel

La lecture à partir d'un fichier en mode texte à accès séquentiel, avec une portion de programme telle que :

```
50 PRINT D$; "READ FICHIER TEXTE"
60 INPUT A$
```

par exemple, se passe d'une manière analogue à celle d'un chargement de programme décrit plus haut. Cependant, seul le secteur contenant la *prochaine rubrique* du fichier en mode texte (tous les caractères à partir du pointeur de position dans le fichier jusqu'au prochain RETURN) est lu et alimenté dans la mémoire tampon, lors de la commande INPUT. Puis les octets correspondants à la rubrique sont transférés en A$. Ce processus est répété si la rubrique s'étend sur plus de un secteur. Chaque INPUT suivant aura pour effet soit le transfert des informations à partir de la mémoire tampon, si elles y sont déjà, soit la lecture effective, et l'alimentation d'un nouveau secteur dans la mémoire tampon. Et ainsi de suite, jusqu'à ce que la dernière rubrique soit lue, ou jusqu'à ce qu'une commande CLOSE soit donnée.

Si le paramètre B (Byte = octet) est utilisé dans la commande READ, le prochain INPUT débutera à l'octet désigné (adresse absolue — le premier octet de la rubrique a le numéro 0). Cet octet peut être situé avant ou après le pointeur de position dans le fichier. Il faut remarquer que, pour utiliser ce paramètre efficacement, il faut parfaitement connaître tous les octets du fichier.

La commande POSITION utilise le paramètre R (pour rubrique Relative), et a pour effet d'*avancer* (jamais de reculer) le pointeur de position dans le fichier du nombre spécifié de rubriques. Chaque fois qu'un fichier est ouvert, le DOS ignore le pointeur de position dans le fichier, et recommence la lecture au début (à moins qu'un paramètre B ne lui donne un ordre différent).

ATTENTION : la commande INPUT ne fonctionne pas exactement de la même manière en BASIC simple et en BASIC étendu. Si certains caractères tels que la virgule ou le point-virgule apparaissent dans la rubrique à alimenter, les caractères suivants peuvent être ignorés ou affectés à des Input à variables multiples (s'il en existe). Pour plus de détails, voir les brochures de BASIC simple ou de BASIC étendu.

La lecture d'un fichier à accès direct

Le processus de lecture est légèrement différent pour la lecture d'un fichier en mode texte à accès direct (l'écriture aussi — voir plus haut dans cette annexe).

Dans un fichier à accès direct, chaque enregistrement comporte le même nombre d'octets. Ce nombre a été spécifié dans le paramètre L (Longueur) de l'ordre OPEN qui a précédé l'écriture du fichier. Lorsque le fichier est de nouveau ouvert avant d'être lu, la même valeur est donnée au paramètre L dans l'ordre OPEN. Pour trouver l'adresse de début d'un enregistrement donné (désigné par le paramètre R dans la commande READ), le DOS utilise la longueur (paramètre L) pour calculer le nombre d'octets occupés par les enregistrements précédents. Puis ce nombre est divisé par 256 ($100), pour déterminer le nombre de secteurs à ignorer avant l'enregistrement. Le DOS se réfère alors à la liste pistes/secteurs, et détermine l'adresse exacte du secteur désiré. Enfin, le DOS alimente ce secteur dans la mémoire tampon, et les octets sont disponibles pour être transférés dans les variables du programme.

Remarquez que le même processus serait appliqué même si le fichier avait été écrit à l'origine comme fichier séquentiel, ou comme fichier à accès direct mais avec une longueur différente. Le DOS calcule aveuglément le secteur et l'octet de début d'enregistrement, en tenant compte de la longueur spécifiée dans l'ordre OPEN qui précède la lecture, indépendamment du paramètre L (s'il y en avait un) utilisé pour l'écriture du même fichier, à l'origine.

Par l'utilisation du paramètre R et du paramètre B dans la commande READ, on peut faire en sorte que le prochain ordre INPUT commence à l'octet désigné à l'intérieur de l'enregistrement spécifié (le premier octet de l'enregistrement a le numéro d'ordre 0). Cet octet peut être situé avant ou après le pointeur de position à l'intérieur de l'enregistrement. Pour l'utiliser efficacement, il faut connaître tous les octets de l'enregistrement.

La commande POSITION a été conçue à l'origine pour les fichiers à accès séquentiel. On peut cependant l'utiliser avec un fichier à accès direct. Le paramètre R (rubrique Relative) fera avancer (jamais reculer) le pointeur de position à l'intérieur de l'enregistrement du nombre de rubriques spécifié. La commande READ utilisée avec le paramètre R (Record, enregistrement, cette fois), déplace le pointeur d'enregistrement vers le 1er octet de l'enregistrement spécifié. L'utilisation de POSITION annule le mode READ (sans restaurer le pointeur de position); une autre commande READ (cette fois *sans paramètre*) ramène au mode READ.

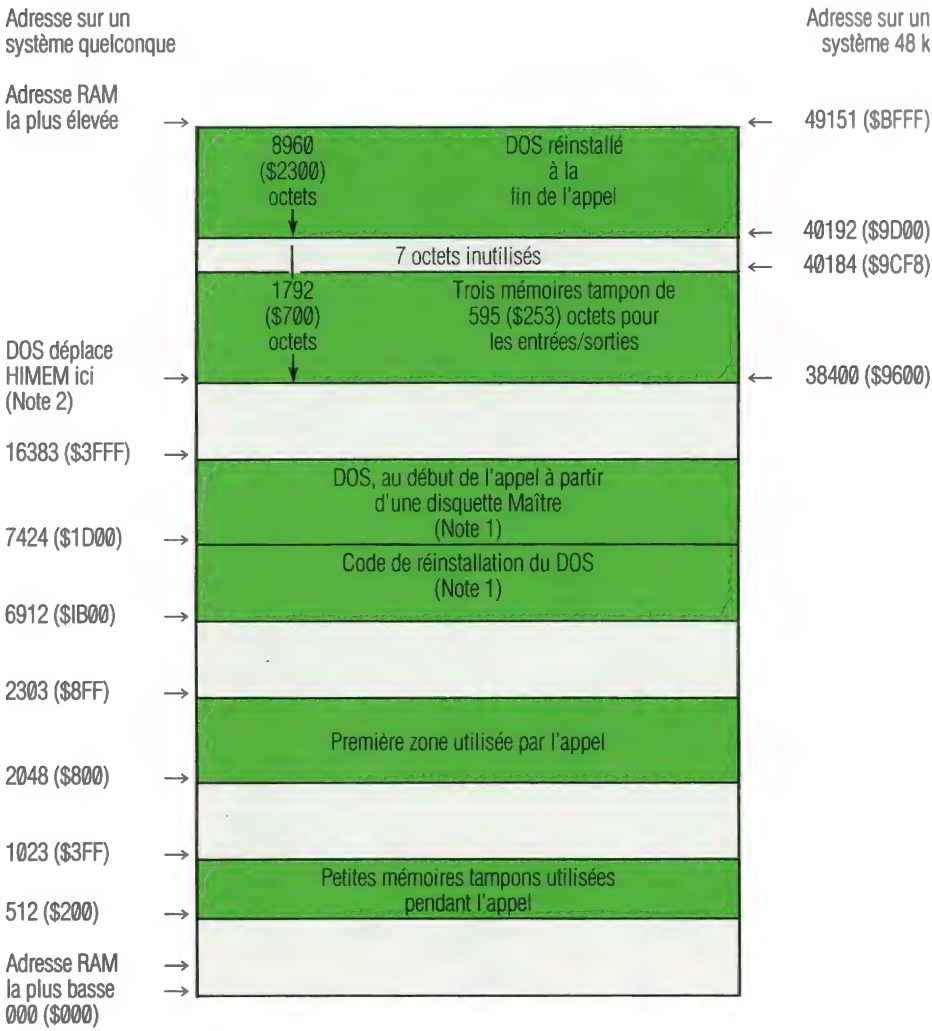
Chaque fois qu'un fichier est ouvert par la commande OPEN, le DOS, ignorant le pointeur de position, repart du début du fichier, à moins qu'un paramètre R (Record) ou B ne lui donne un ordre contraire.

ATTENTION : le DOS n'enregistre aucune information concernant la structure, le format, la longueur des enregistrements ou des rubriques de vos fichiers en mode texte. Pour utiliser correctement vos fichiers en accès direct, il est indispensable que vous teniez une documentation complète sur ces points. Vous pouvez la mémoriser par écrit, ou mieux encore, dans l'enregistrement de numéro d'ordre 0 de votre fichier.

ANNEXE D
Schéma de la mémoire

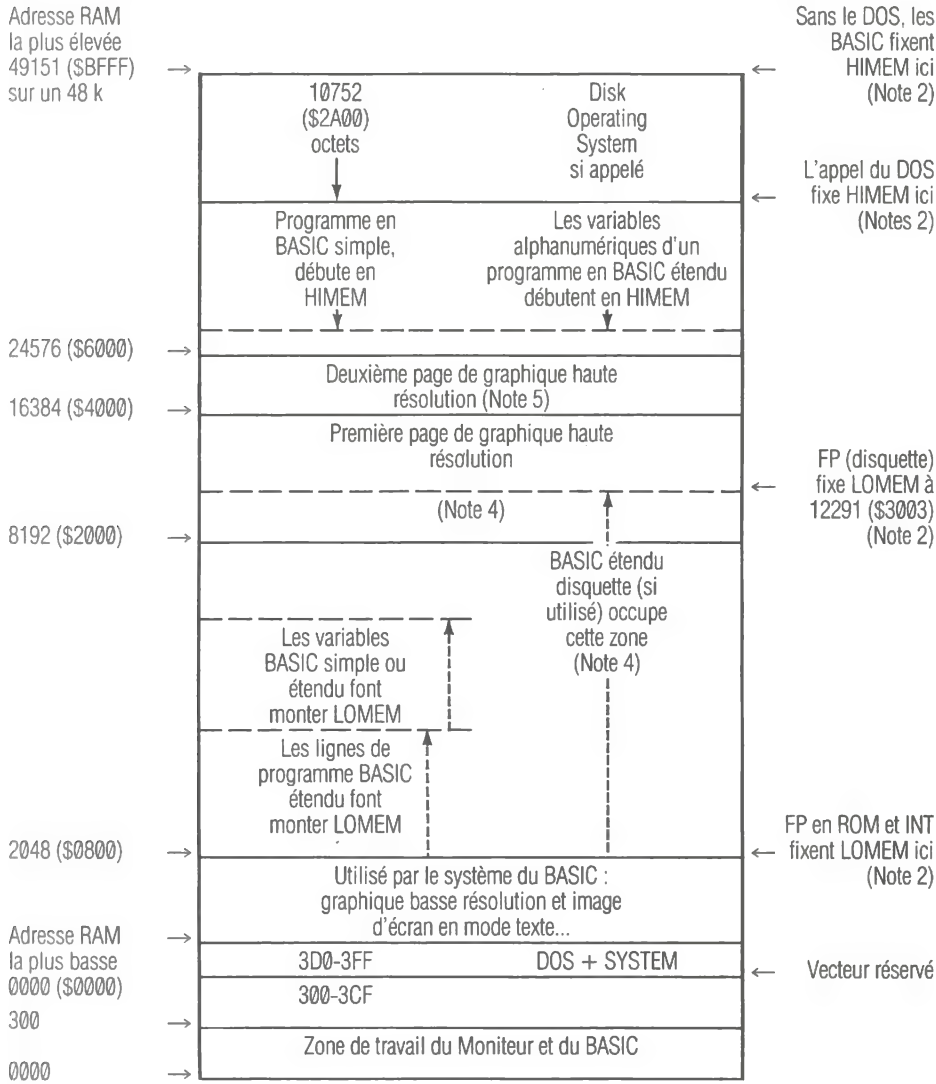
132 Zones de mémoire utilisées par le DOS lors de l'appel
133 Zones de mémoire utilisées par le DOS et le BASIC (simple ou étendu)
134 Valeurs de HIMEM fixées par l'appel du DOS

Zones de mémoire utilisées par le DOS lors de l'appel



Note 1. Cette zone mémoire n'est pas utilisée lorsque le DOS est appelé à partir d'une disquette secondaire; dans ce cas, le DOS est immédiatement placé *sous l'adresse haute de la mémoire du système sur lequel la disquette secondaire a été initialisée,* même si celle-ci ne concorde pas avec le système utilisé lors de l'appel.

Zones de mémoire utilisées par le DOS et le BASIC (simple ou étendu)



Note 2 : Si votre système est en BASIC simple, l'adresse de HIMEM peut être trouvée (octet unitaire d'abord, puis octet x 256) en adresse 76-77 ($4C-$4D).
Si votre système est en BASIC étendu, l'adresse de HIMEM est en 115-116 ($73-$74), selon le même format. Voir ci-dessous en table 2 la valeur de HIMEM après l'appel du DOS, sur les différentes configurations de mémoire. Le fait d'augmenter le nombre de mémoires tampon, par l'utilisation de MAXFILES, fait descendre l'adresse de HIMEM de 595 octets, pour chaque mémoire tampon ajoutée. Pour connaître les adresses des autres pointeurs du BASIC étendu, voir la brochure concernée, Annexe 1.

Table 2 : Valeur de HIMEM fixée par l'appel du DOS

Lorsque le DOS est appelé, l'adresse de HIMEM est fixée en fonction de la mémoire disponible dans le système.

Voir le tableau ci-dessous :

Taille mémoire	Adresse RAM		HIMEM fixé par DOS	
	Décimale	Hexa	Décimale	Hexa
16 K	16383	$3FFF	5632	$1600
20 K	20479	$4FFF	9728	$2600
24 K	24575	$5FFF	13824	$3600
32 K	32767	$7FFF	22016	$5600
36 K	36863	$8FFF	26112	$6600
48 K	49151	$BFFF	–27136	$9600 (Note 3)

Note 3. Le nombre –27136 pourrait s'écrire 38400, mais le BASIC entier ne permet pas la manipulation de nombres supérieurs à 32767. C'est pourquoi, en BASIC entier, les adresses supérieures à 32767 sont exprimées en leur équivalent négatif. L'équivalent négatif de toute adresse décimale positive n est : n - 65536.

Note 4. Lors de l'utilisation de graphiques haute résolution, la page 1 affectée à l'image mémoire du graphique occupe les mémoires 8192 à 16383. A moins que le DOS n'ait fixé l'adresse du HIMEM à une valeur supérieure à 16383, une tentative d'utilisation de graphique haute résolution alors que le système se trouve sous DOS aboutira à « écraser » une partie du DOS. Ceci a pour conséquence l'obligation de disposer de 32 K RAM minimum, si l'on désire utiliser simultanément le DOS et les graphiques haute résolution.

Si l'on utilise le BASIC étendu à partir de la disquette, une tentative d'utilisation de la page 1 de graphique haute résolution « écraserait » le BASIC étendu. Avec le BASIC étendu sur disquette, seule la page 2 de graphique haute résolution peut être utilisée. Encore faut-il disposer de 36 K de mémoire minimum.

Note 5. L'utilisation de la page 2 de graphique haute résolution occupe les mémoires 16384 à 24575. A moins que le DOS n'ait fixé l'adresse de HIMEM à une valeur supérieure à 25675, une tentative d'utilisation de la page 2 de graphique haute résolution aboutira à écraser le DOS. Pour utiliser le DOS et la page 2 haute résolution, il faut disposer de 36 K minimum de mémoire RAM.

ANNEXE E
Points d'entrée du DOS

136 Points d'entrée du DOS
137 Schéma des circuits de l'interface du DISK II
138 Schéma des circuits de la carte analogique du DISK II

Points d'entrée du DOS

Les routines pour reconnecter le DOS (si la page 3 est "écrasée" sont les suivantes :

Taille mémoire	Adresse décimale (CALL)	Adresse hexadécimale (G)
48K	–25153	$9DBF
32K	23999	$5DBF
16K	7615	$1DBF

L'adresse hexadécimale ci-dessus correspondante à la taille mémoire disponible peut être retrouvée par la commande Moniteur 3DØL, si la page 3 n'a pas été écrasée.

Adresses contenant l'adresse de début et la longueur d'un programme chargé par BLOAD :

Taille mémoire	Adresse de début (octet unitaire)		Longueur du programme (octet unitaire)	
	Décimale	Hexadécimale	Décimale	Hexadécimale
49K	43634	$AA72	43616	$AA60
32K	27250	$6A72	27232	$6A60
16K	10866	$2A72	10848	$2A60

Pour visualiser l'adresse de début ou la longueur d'un programme chargé par BLOAD, il suffit de taper :

PRINT PEEK (octet de faible poids) + PEEK (octet de faible poids +1) * 256

Vous trouverez ci-dessous un programme permettant de trouver l'adresse de début, et la longueur du dernier programme chargé par BLOAD, quelle que soit la taille de la mémoire :

```
  5 REM BLOAD FINDER
  7 H = 38400 : REM HIMEM FIXE PAR L'APPEL DU DOS
  8 T = 49152 : REM ADRESSE LA PLUS ELEVEE
 10 D$ = CHR$ (4) : REM CTRL-D
 20  PRINT D$; "BSAVE FOO, A$7777, L$77"
 30  PRINT D$; "BLOAD FOO"
 40  PRINT D$; "DELETE FOO"
 50  FOR I = H + 1792 TO T
 60  IF PEEK (I) <> 119 OR PEEK
     (I + 1) <> 119 THEN NEXT I
 70  PRINT "ADRESSE DE L'ADRESSE DE DEPART: ";I;",";I+1
 80  FOR I = H + 1792 TO T
 90  IF PEEK (I) <> 119 OR PEEK
     (I + 1) <> Ø THEN NEXT I
100  PRINT "ADRESSE DE LA LONGUEUR : "
     ;I;",";I+1
```

Les valeurs de H et de T (lignes 7 et 8) sont données pour un système 48K. L'Annexe D donne les valeurs pour toutes les configurations. Ce programme prend environ deux minutes pour donner les adresses désirées.

Schéma des circuits de l'interface du DISK II

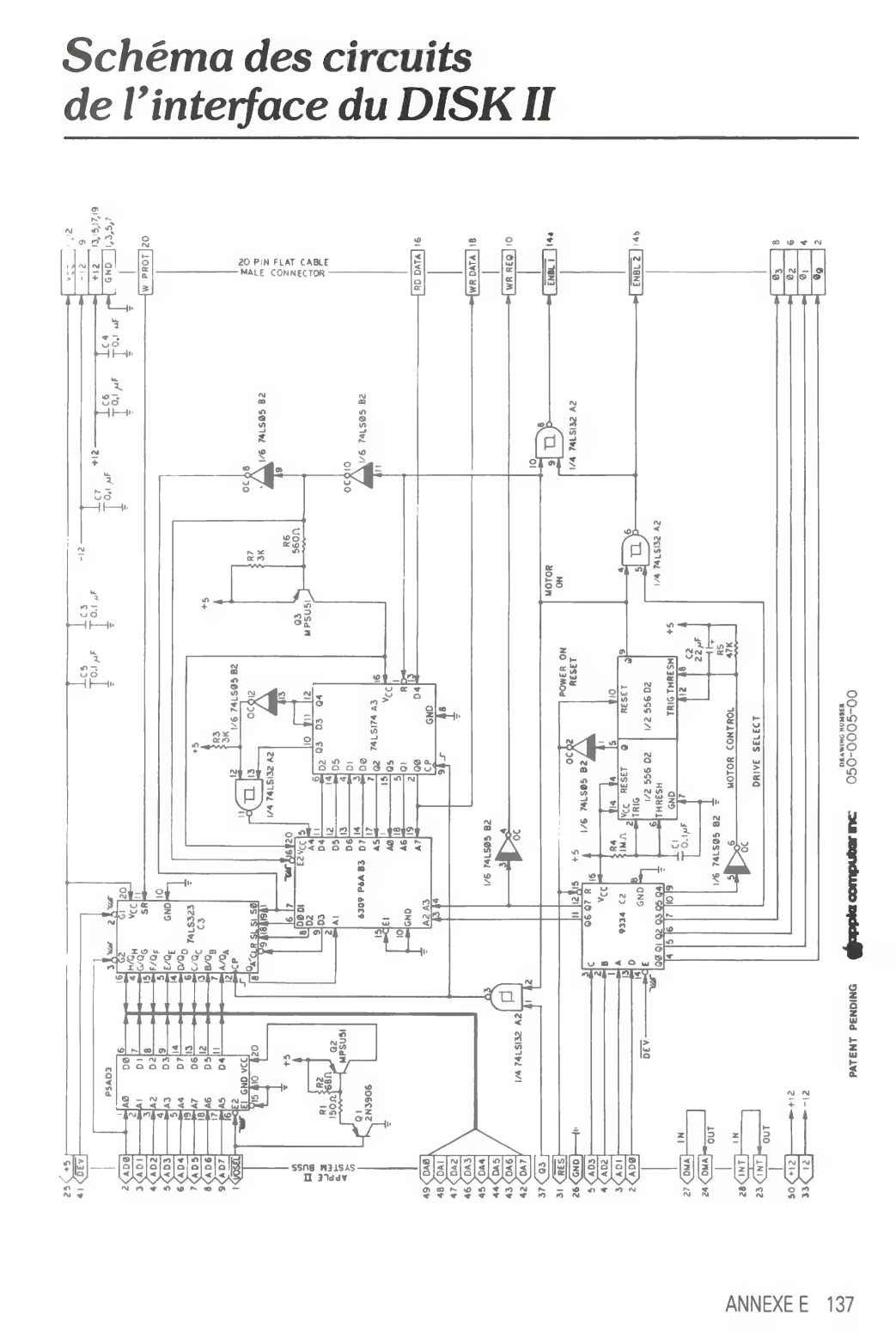
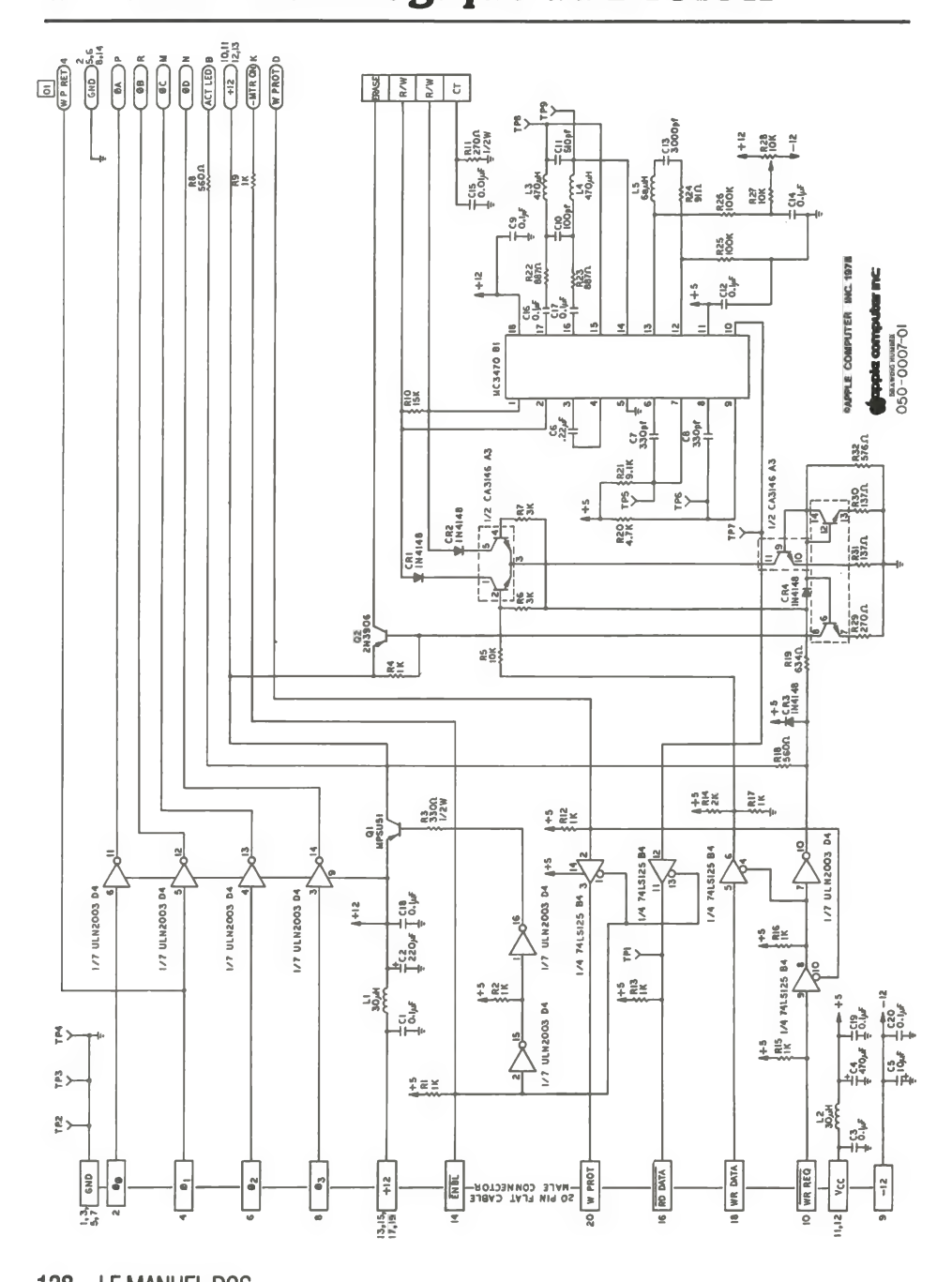


Schéma des circuits de la carte analogique du DISK II



ANNEXE F
Sommaire des commandes du DOS

140 Notation
142 Les noms de fichiers
143 Les commandes de service
147 Les commandes initiales
149 Les commandes pour les fichiers en mode texte à accès séquentiel
152 Les commandes pour les fichiers en mode texte à accès direct
153 Les commandes pour les fichiers en mode binaire

Les commandes du DOS sont regroupées en cinq catégories dans cette annexe :

Commandes de service

INIT	RENAME	VERIFY
CATALOG	DELETE	MON
SAVE	LOCK	NOMON
LOAD	UNLOCK	MAXFILES
RUN		

Commandes initiales

FP	PR#	CHAIN
INT	IN#	

Commandes pour les fichiers en mode texte à accès séquentiel

OPEN	APPEND
CLOSE	POSITION
READ	EXEC
WRITE	

Commandes pour les fichiers en mode texte à accès direct

OPEN	READ
CLOSE	WRITE

Commandes pour les fichiers en mode binaire

BSAVE
BLOAD
BRUN

Les procédures utilisées avec le DOS, y compris le recouvrement de programme (CHAIN) en BASIC étendu, sont résumées dans l'Annexe G. La notation utilisée dans le Manuel et les Annexes est décrite ci-après :

Notation

La syntaxe concerne la structure d'une commande. Une notation simple a été adoptée pour décrire la syntaxe de chaque commande.

– Les éléments entre parenthèses sont optionnels. Ils sont parfois appelés des paramètres. Toutes les commandes n'admettent pas tous les paramètres, mais les paramètres admis pour une commande peuvent apparaître dans un ordre quelconque, à moins qu'un ordre ne soit spécifiquement imposé.

– Si une commande du DOS porte sur un nom de fichier, celui-ci doit suivre immédiatement le mot de commande : le premier élément suivant la commande sera de toute façon pris pour un nom de fichier. Si un paramète suit le nom de fichier, il doit être séparé de celui-ci par une virgule.

– Le mot {CTRL} suivi d'un tiret signifie qu'il convient de maintenir la touche contrôle CTRL enfoncée, puis de taper la lettre qui suit le tiret. Exemple : CTRL-D.

– {RETURN} signifie qu'il convient de frapper la touche RETURN. Cette opération est nécessaire après chaque commande, mais n'est pas chaque fois rappelée.

– {RESET} signifie qu'il convient de frapper la touche RESET.

- {ESC} signifie qu'il convient de frapper la touche escape ESC.

- Les éléments en lettres majuscules doivent être strictement recopiés.
Par contre, les lettres minuscules remplacent des éléments qui doivent être fournis par l'utilisateur.

f nom de fichier. De un à trente caractères. Tous les caractères du clavier, sauf la virgule, sont autorisés dans les noms de fichier. Le premier caractère doit être une lettre de l'alphabet.

g un autre nom de fichier.

s un numéro de port, ou de connecteur d'interface. En fait, s spécifie le numéro de connecteur dans lequel la carte d'interface de la disquette a été insérée (généralement 6).
Par défaut, c'est le connecteur utilisé lors de l'appel du DOS qui est concerné, ou la dernière valeur affectée à s.
s peut avoir une valeur variant de 1 à 7. Exemple : 7/2.

ATTENTION : si s se réfère à un connecteur qui ne contient pas d'interface de disquette, le système s'arrêtera, et le programme en mémoire pourrait être perdu (voir l'Annexe B -I/O ERROR — à ce sujet).

v numéro de volume d'une disquette. Initialement, v prend par défaut le numéro de volume de la disquette à partir de laquelle le DOS a été appelé, ou la dernière valeur affectée à v, ou une valeur affectée implicitement par une commande CATALOG. La valeur de v peut varier de Ø à 254.
Note : le numéro de volume d'une disquette ne peut pas être Ø. Une valeur Ø pour un numéro de volume, ou pas de valeur du tout, dans une commande, indique simplement au DOS qu'il convient d'utiliser le numéro de volume de la disquette en cours.

d numéro de lecteur (1 ou 2). La valeur initiale par défaut est 1. La valeur par défaut est la dernière valeur donnée pour d.

p numéro de position. p est utilisé avec le paramètre R dans les commandes POSITION et EXEC pour les fichiers en mode texte à accès séquentiel. Le numéro de position indique une rubrique dont la position dans le fichier est à p rubriques en avant de la position du pointeur du fichier.
La valeur par défaut de p est Ø, ce qui ne modifie pas la position du pointeur de fichier.
Note : La commande EXEC initialise toujours le pointeur au début du fichier. C'est pourquoi le paramètre p se calcule toujours à partir de la position Ø lorsqu'il est utilisé avec la commande EXEC. La valeur de p peut varier de Ø à 32767.

r numéro d'enregistrement. Ce paramètre est utilisé avec les commandes READ et WRITE pour les fichiers en mode texte à accfès direct. La valeur par défaut est la dernière valeur de r spécifiée pour ce fichier. Le paramètre r désigne un numéro d'ordre d'enregistrement — en position absolue — à l'intérieur d'un fichier. Sa valeur peut varier de Ø à 32767.

a adresse en mémoire RAM. Ce paramètre est utilisé avec la commande BSAVE. Le paramètre a spécifie l'adresse de début pour le chargement ou la sauvegarde d'un fichier en mode binaire. Si le paramètre a n'est pas utilisé pour le chargement (BLOAD), sa valeur par défaut est celle qui a été utilisée lors de la sauvegarde BSAVE). La valeur de a peut varier de Ø à 65535.

b numéro d'octet (byte). La valeur par défaut de b est Ø. Dans un fichier à accès séquentiel, b désigne un octet, en position absolue, à l'intérieur d'un fichier. Dans un fichier à accès direct, b désigne un octet, en position absolue, à l'intérieur de l'enregistrement déterminé par le

paramètre r. La valeur de b peut varier de 0 à 32767. Dans la pratique, la valeur de b varie de 0 à la longueur d'un enregistrement (accès direct) ou de 0 à la longueur d'un fichier (accès séquentiel).

j est une indication de longueur. La valeur par défaut de j est 1.
Lorsque l'on ouvre un fichier en accès direct par une commande OPEN, j est indispensable pour donner une valeur au paramètre L et indiquer la longueur de l'enregistrement.
Lorsque la commande BSAVE est utilisée, j est indispensable, et indique le nombre d'octets, à partir de l'adresse a, à mémoriser sur disquette.
La valeur de j peut varier de 0 à 32767.

Toute constante numérique dans une commande DOS peut être donnée en notation décimale, ou bien en notation hexadécimale, en la faisant précéder du signe $.

Les noms de fichiers

Les noms de fichiers peuvent comporter jusqu'à 30 caractères. Le premier caractère doit être une lettre. Le nom ne peut pas contenir une virgule, un caractère CTRL-M ou un retour chariot (RETURN), qui est utilisé pour signifier la fin de la commande. Les espaces qui précèdent le premier caractère significatif sont ignorés. Tous les caractères au-delà du 30 sont ignorés.

Si des touches spéciales sont utilisées dans les noms de fichiers, telles que ESC, flèche droite, flèche gauche et certains caractères de contrôle (CTRL-C, CTRL-H), les conséquences peuvent être surprenantes.

Si un nom de fichier contient des caractères de contrôle, ils ne seront pas affichés, mais il sera nécessaire de les taper pour utiliser ou pour supprimer le fichier.

Le programme suivant, écrit en BASIC étendu, permet de trouver des caractères de contrôle dissimulés dans un nom de fichier, sauf CTRL-M (RETURN), ESC, CTRL-H (flèche gauche) et CTRL-U (flèche droite).

```
10  DATA 201,141,240,21,201,136
20  DATA 240,17,201,128,144,13
30  DATA 201,160,176,9,72,132
40  DATA 53,56,233,64,76,249
50  DATA 253,76,240,253
60  FOR I = 768 TO 768 + 27
70  READ V : POKE I,V : NEXT I
80  POKE 54,0 : POKE 55,3
90  CALL 1002
```

Si vous pensez avoir tapé par erreur un caractère de contrôle dans un nom de fichier, tapez ce programme, sauvegardez-le puis exécutez-le. Le caractère d'appel] du BASIC étendu sera affiché. Tapez CATALOG.

Si les noms de programmes / fichiers contiennent des caractères de contrôle, ils apparaîtront dans la liste du catalogue, en mode inversé. Les caractères de contrôle dans les programmes peuvent aussi être trouvés de cette manière. Pour revenir à un affichage normal, il suffit de donner l'ordre PR # 0.

→ INIT f (,Vv) (,Ss) (,Dd)

Exemple :

INIT HELLO,V18

Le paramètre v assigne un numéro de volume à la disquette en cours d'initialisation. Pour plus de détails, voir le Chapitre 2 et l'Annexe G.

→ CATALOG (s.Ss) (,Dd)

Exemple :

CATALOG

Cette commande affiche à l'écran le numéro de volume de la disquette et la liste des fichiers et programmes qui y sont mémorisés. Si le paramètre v (,Vv) est utilisé dans cette commande, il est ignoré.
Précédant chaque nom de fichier ou de programme, se trouve un indicateur de type et le nombre de secteurs occupés sur la disquette. Les types de fichiers ou de programmes sont :

I Programme en BASIC simple, écrit par SAVE

A Programme en BASIC étendu, écrit par SZVE

T Fichier en mode texte cré par OPEN et écrit par WRITE

B Image mémoire binaire, créé par BSAVE

Un astérisque à côté d'un code type signifie qu'un fichier ou un programme est verrouillé (LOCK).

Au total, 496 secteurs sont disponibles pour l'utilisateur. Chaque secteur permet de mémoriser 256 octets. La taille minimum d'un fichier est 1 secteur, pour un fichier vide en mode texte (en fait ce secteur est occupé par la liste piste / secteurs du fichier, elle-même vide). Un programme vide, de type I, A ou B prend 2 secteurs (1 pour la liste pistes / secteurs et 1 pour le premier secteur du programme, qui contient l'indication de la longueur du programme. Voir l'annexe C pour une information plus complète).

Remarquez que si la taille d'un fichier ou d'un programme excède 255 secteurs, le compteur de secteurs du catalogue repart de 000. Ceci n'affecte aucunement l'utilisation du fichier, mais peut donner une fausse impression quant à la place encore disponible sur la disquette.

→ SAVE f (,Ss) (,Dd) (,Vv)

Exemple :

SAVE COLOR DEMOS,V56

S'il n'y a pas de programme de ce nom dans le lecteur spécifié (ou dans le lecteur par défaut), un fichier est créé portant ce nom, et le programme en mémoire est mémorisé dans ce fichier. Si la disquette contient déjà un fichier de ce nom, mais dont le type ou le langage est différent, le message

FILE TYPE MISMATCH

est affiché.

Si la disquette contient déjà un fichier de ce nom, de même type ou langage, le nouveau programme le remplace en totalité, sans que le DOS ne donne d'avertissement.

→ LOAD f (,Ss) (,Dd) (,Vv)

Exemple :

LOAD DOW JONES,V19,D1

Le DOS recherche un programme BASIC simple ou étendu, de nom f, sur la disquette du lecteur spécifié (ou le lecteur par défaut). Si les numéros de volume concordent, et si le nom de fichier existe, le programme sera chargé dans la mémoire du micro-ordinateur.

La commande LOAD ferme tout fichier en mode texte qui serait ouvert, met le système dans le langage du programme chargé, et efface tout programme qui se trouve préalablement en mémoire.

Si le fichier f est un programme en BASIC étendu, et que le BASIC étendu n'est pas en mémoire, ou disponible en ROM, il sera chargé à partir de la disquette, préalablement au chargement du programme. Si le BASIC étendu n'est ni en ROM ni sur la disquette, le message

LANGUAGE NOT AVAILABLE

(langage non disponible) sera affiché.

Si une commande LOAD est donnée sans aucun paramètre, le système tentera de charger un programme à partir du lecteur de cassettes.

→ RUN f (,Ss) (,Dd) (,Vv)

Exemple :

RUN ANNUITY,D2

a pour effet de charger le fichier f à partir du lecteur de disquette spécifié (ou de lecteur par défaut), puis d'exécuter le programme.

Si la commande RUN est tapée sans paramètre, c'est le programe qui est en mémoire centrale qui est exécuté.

→ RENAME f,g (,Ss) (,Dd) (,Vv)

Exemple :

RENAME SEPERATE,SEPARATE,S4,D1,V0

a pour effet de repérer le fichier nommé f sur la disquette du lecteur spécifié (ou du lecteur par défaut), et de lui affecter dorénavant le nom g. Le contenu du fichier n'est pas modifié. Si le fichier f était ouvert, il est fermé.

ATTENTION : la commande RENAME ne contrôle pas que le nom g n'existe pas déjà sur la disquette. On pourrait donc, avec RENAME, avoir plusieurs fichiers comportant le même nom sur la disquette. Une situation à éviter absolument !

→ DELETE f, (,Ss) (,Dd) (,Vv)

Exemple :

DELETE TEST

supprime le fichier nommé f de la disquette dans le lecteur spécifié (ou dans le lecteur par défaut). Si le fichier f était ouvert, cette commande le ferme. Voir l'Annexe C pour une information plus complète sur l'opération de suppression.

Si le fichier nommé f n'est pas sur la disquette, le message

FILE NOT FOUND

fichier non trouvé, est affiché. Pour éviter que ce message n'interrompe votre programme, vous pouvez ouvrir le fichier (OPEN) avant de le supprimer (DELETE).

→ LOCK f (,Ss) (,Dd) (,Vv)

Exemple :

LOCK LOVE LETTERS,V31

Cette commande permet de protéger le fichier f sur la disquette du lecteur spécifié (ou du lecteur par défaut) en le verrouillant. Un fichier verrouillé est signalé par la présence d'un astérisque précédant son nom dans le catalogue.

→ UNLOCK f (,Ss) (,Dd) (,Vv)

Exemple :

UNLOCK RECIPES,V31,D2

permet de déverrouiller le fichier nommé f.

→ VERIFY f (,Ss) (,Dd) (,Vv)

Exemple :

VERIFY SAM

effectue un conrôle interne des informations mémorisées sur la disquette : lors de l'écriture par SAVE, BSAVE ou WRITE, le DOS calcule un octet de contrôle en fonction du contenu de chaque mémoire tampon, et mémorise cet octet sur le secteur de la disquette; la commande VERIFY calcule à nouveau l'octet de contrôle mémorisé lors de l'écriture. Si aucune inégalité n'est détectée, aucun message n'est affiché par la commande VERIFY, et l'on peut raisonnablement penser que les informations sont correctement mémorisées sur la disquette. Si une inégalité est détectée, le message :

I/O ERROR

est affiché.

N'importe quel type de fichier peut être contrôlé par une commande VERIFY.

→ MON (C) (,I) (,O)

Exemples :

MON O
MON C,I,O

Toutes les commandes disquettes et toutes les informations circulant entre le micro-ordinateur et les disquettes ne sont pas normalement affichées à l'écran. Cette commande permet d'en visualiser tout ou partie. Il s'agit en fait d'un outil très utile pour la mise au point des programmes. Si « C » est spécifié, les *commandes du DOS* sont affichées. Si « I » est spécifié, toutes les informations envoyées de la disquette au micro-ordinateur sont affichées (Input - entrée). Si "O" est spécifié, toutes les informations envoyées du micro-ordinateur à la disquette sont affichées (Output - sortie).

Il est nécessaire qu'au moins l'un des trois paramètres soit présent, faute de quoi la commande MON est ignorée.

Les paramètres peuvent être donnés dans un ordre quelconque, séparés par une virgule. Les paramètres n'apparaissent que dans les ordres MON et NOMON.

Note : MON reste en fonction jusqu'à
- ce qu'une commande NOMON soit donnée;
- ce qu'un changement de langage (FP ou INT) soit effectué;
- un nouvel appel du DOS;
- un retour au BASIC par 3DØG.

Le fait d'exécuter un nouveau programme (RUN) ne mettra pas MON hors fonction.

→ NOMON (C) (,I) (,O)

Exemples :

NOMON C
NOMON I,C

La commande MON permet de visualiser les commandes disquettes et les informations circulant entre le micro-ordinateur et les disquettes, qui ne sont pas normalement affichées par le DOS. La commande NOMON permet d'annuler tout ou partie des effets d'une commande MON.

NOMON C,I,O

Remet le système dans son état normal.

Si « C » est spécifié, les commandes du DOS ne sont pas affichées. Si « I » est spécifié, les informations envoyées de la disquette au micro-ordinateur (Input - entrée) ne sont pas affichées. Si « O » est spécifié, les informations envoyées du micro-ordinateur à la disquette (Output - sortie) ne sont pas affichées.

Il est nécessaire qu'au moins l'un des trois paramètres soit présent, faute de quoi la commande est ignorée. Les paramètres peuvent être donnés dans un ordre quelconque, séparés par une virgule. Ces paramètres n'apparaissent que dans les ordres MON et NOMON.

→ MAXFILES n

Exemple :

MAXFILES 6

n est un entier dont la valeur peut varier de 1 à 16. Il indique le nombre maximum de fichiers pouvant être simultanément ouverts. Lorsque la commande MAXFILES s'exécute, 595 octets (mémoire tampon) sont réservés pour chaque fichier.

Lorsque le DOS est appelé, la valeur par défaut de n est 3. Le nombre de fichiers qui peuvent être simultanément ouverts est donc de 3, sans commande MAXFILES, 1785 octets étant pris pour les mémoires tampon. Toutes les commandes du DOS, sauf PR #, IN # et MAXFILES utilisent une mémoire tampon. C'est pourquoi, si vous aviez donné un ordre MAXFILES 1, et qu'un fichier se trouve ouvert, une tentative pour exécuter une commande du DOS, comme CATALOG, par exemple, donnerait le message

NO BUFFERS AVAILABLE

(pas de mémoire tampon disponible).

ATTENTION : la commande MAXFILES déplace l'adresse de HIMEM. Les programmes (BASIC simple) ou les variables (BASIC étendu) en mémoire peuvent donc être détruits. C'est pourquoi la commande MAXFILES doit être donnée avant le chargement et l'exécution d'un programme. Pour l'utilisation de la commande MAXFILES dans un programme, voir les Chapitres 5 et 7.

Les commandes initiales

FP (,Ss) (,Dd) (,Vv)

Exemple :

FP,D2

Cette commande met le système en BASIC étendu. Tout programme (BASIC simple ou BASIC étendu) en mémoire est détruit. Si votre micro-ordinateur comporte le BASIC étendu en ROM, ou la carte BASIC étendu, le DOS utilisera cette source, indépendamment de la position de l'interrupteur de la carte. Sinon, le DOS tente de charger à partir de la disquette, puis d'exécuter le programme nommé APPLESOFT. Pour écrire le programme APPLESOFT sur une disquette normalement initialisée, il suffit de :

– charger (LOAD) le programme APPLESOFT à partir d'une disquette Maître;
– le sauvegarder sur la disquette, sans le lister (LIST) ou l'exécuter (RUN) avant. Le nom pour la sauvegarde est obligatoirement APPLESOFT.

ATTENTION : pour passer en BASIC étendu, ne tapez pas RUN APPLESOFT. Bien que ce procédé donnerait l'apparence de fonctionner, il n'en serait rien.

INT

Exemple :

INT

Cette commande met le système en BASIC simple. Tout programme en mémoire (BASIC simple ou étendu) est perdu.

CTRL-D

Exemple :

```
10  D$ = CHR$(4)
20  PRINT D$; "WRITE CHESS"
```

Tout caractère affiché ou imprimé (PRINT) par le micro-ordinateur est, dans un premier temps, examiné par le DOS avant d'être envoyé vers une unité de sortie. Si un retour chariot (RETURN) est suivi d'un caractère CTRL-D, il s'agit d'un message au DOS pour lui signifier que tous les caractères à venir (jusqu'au prochain RETURN) sont une commande du DOS. La plupart des commandes du DOS peuvent être utilisées à partir d'un programme (BASIC simple ou étendu). Pour le faire, il suffit de donner un ordre PRINT, suivi d'un CTRL-D, suivi d'une commande du DOS.

Il est recommandé de définir une variable alphanumérique contenant le caractère CTRL-D, puis de l'utiliser comme montré dans l'exemple ci-dessus. Remarquez que la fonction CHR$ (indisponible en BASIC simple) a été utilisée pour définir le CTRL-D. En BASIC simple, il suffit d'écrire le CTRL-D entre deux guillemets : D$ = "". Le caractère CTRL-D n'apparaît pas à l'écran.

Voir l'Annexe G pour une plus ample information sur l'utilisation du DOS à partir d'un programme.

PR # s

a pour effet de diriger toutes les sorties vers l'unité contrôlée par le connecteur numéro s, au lieu de l'écran de visualisation. La commande PR # 0 ramène à la sortie écran. Si cette commande est utilisée dans un programme, elle doit être traitée comme une commande du DOS :

```
10  D$ = "" : REM CTRL-D
20  PRINT D$; "PR# 1"
```

Si le connecteur numéro s ne comporte pas de carte d'interface, le système peut se «planter», et le seul moyen de reprendre la main est de frapper la touche RESET.

IN # s

a pour effet d'accepter toutes les entrées à partir de l'unité contrôlée par le connecteur numéro s, au lieu du clavier. La commande IN # 0 ramène à l'entrée clavier. Si la commande est utilisée dans un programme, elle doit être traitée comme une commande du DOS :

```
10  D$ = "" : REM CTRL-D
20  PRINT D$; "IN# 1"
```

Si le connecteur numéro s ne comporte pas de carte d'interface, le système peut se "planter", et le seul moyen de reprendre la main est de frapper la touche RESET.

CHAIN f (,Ss) (,Dd) (,Vv)

Exemple :

```
CHAIN PARTIE DEUX,D1,S7,V0
```

Utilisée à partir d'un programme en BASIC simple, cette commande a pour effet de charger et d'exécuter le programme f (sur la disquette spécifiée, ou sur la disquette par défaut), sans perdre les variables du premier programme. Ceci a notamment comme avantage la possibilité pour le programme f de disposer des résultats du programme antérieur, et de passer des informations à un programme suivant. On ne peut réaliser de recouvrement de programme en BASIC étendu avec la commande CHAIN. Voir le Chapitre 10 et l'Annexe G sur ce point.

Les commandes pour les fichiers en mode texte à accès séquentiel

OPEN f (.Ss) (.Dd) (.Vv)

Exemple :

OPEN SESAME.D2

cette commande alloue une mémoire tampon de 595 octets pour le fichier en mode texte f, et prépare le DOS à écrire ou à lire à partir du *début* du fichier. Cette commande est utilisée en liaison avec les commandes WRITE et READ pour créer et consulter des fichiers en accès séquentiel.

S'il n'y a pas de fichier f sur la disquette du lecteur spécifié (ou du lecteur par défaut), un fichier f est créé. Si un fichier nommé f est déjà ouvert, celui-ci est d'abord fermé (CLOSE), avant d'être à nouveau ouvert.

CLOSE f

Exemple :

CLOSE WINDOW

S'il s'agit d'un fichier en écriture, la commande CLOSE provoque l'écriture des caractères en mémoire tampon. La mémoire tampon du fichier f est désallouée. Si la commande CLOSE est utilisée sans nom de fichier, tous les fichiers sont fermés, sauf éventuellement le fichier en EXEC. Rappelons qu'il ne peut y avoir qu'une seule procédure cataloguée (EXEC) en fonctionnement à un moment donné. Lorsqu'une seconde procédure cataloguée est ouverte, la première est automatiquement fermée.

Si un programme en cours d'exécution, contrôlé par une procédure cataloguée, est interrompu par un CTRL-C, il est recommandé de taper CLOSE, pour éviter la perte éventuelle d'enregistrements.

ATTENTION : tout fichier qui a été ouvert (OPEN) doit être fermé (CLOSE), faute de quoi le contenu de la mémoire tampon peut être perdu (perte éventuelle des derniers enregistrements écrits).

WRITE f (.Bb)

Exemple :

WRITE ADDRESS.DATA

Après cette commande, les ordres PRINT dirigent les caractères vers le fichier spécifié, au lieu de les diriger vers l'écran de visualisation. Avec l'utilisation du paramètre B (Byte - octet), l'écriture débute au b ième octet du fichier (voir Chapitre 6). La commande WRITE reste en fonctionnement jusqu'à

- une autre commande DOS;
- un ordre INPUT;
- une commande DOS nulle telle que PRINT CTRL-D.

La commande WRITE ne peut fonctionner qu'en mode différé, c'est-à-dire à partir d'un programme.

Il convient de remarquer que lorsque la commande WRITE est en fonctionnement, tous les caractères qui seraient normalement affichés à l'écran sont écrits dans le fichier, y compris les éventueles messages d'erreur, points d'interrogation associés à un INPUT, et autres caractères indésirables!

READ f (,Bb)

Après cette commande, les caractères sont acceptés par INPUT (et GET en BASIC étendu) à partir du fichier séquentiel f, au lieu du clavier. Avec le paramètre B, la lecture débute au b ième octet du fichier (voir Chapitre 6). L'ordre INPUT accepte les caractères lus dans le fichier séquentiel, rubrique par rubrique. Une rubrique est constituée par une suite de caractères (1 à 32767 caractères) et se termine par un retour chariot (RETURN). Dans la pratique, la longueur des rubriques est limitée à 255 caractères, à cause de la limite de longueur des chaînes de caractères et aussi de la taille de la mémoire tampon.

La commande READ reste en fonctionnement jusqu'à

- une autre commande DOS;
- une commande DOS nulle telle que PRINT CTRL-D.

La commande READ ne peut fonctionner qu'en mode différé (c'est-à-dire à partir d'un programme).

APPEND f (,Ss) (,Dd) (,Vv)

Exemple :

```
APPEND MORE INFO
```

Cette commande ouvre le fichier f, mais positionne le pointeur de fichier à la fin du fichier. Après cette commande, le prochain caractère écrit dans le fichier suivra le dernier caractère précédemment écrit dans le fichier. La commande APPEND doit être suivie de la commande WRITE pour le fichier concerné (APPEND ne doit pas être suivi d'une commande OPEN pour le même fichier, car celle-ci repositionnerait le pointeur du fichier).

POSITION f (,Rp)

Exemple :

```
POSITION ADDRESS.DATA,R277
```

POSITION place le pointeur de fichier au début de la p ième rubrique *suivant* la rubrique en cours. Toute commande READ ou WRITE suivante débutera à cette position dans le fichier f.

Remarquez qu'il s'agit d'une position relative, et non absolue, puisqu'il faut la calculer à partir de la position en cours. POSITION ne peut que faire *avancer* le pointeur de fichier, et non le faire *reculer* par rapport à la position en cours.

En fait, la commande POSITION consiste en une exploration du fichier, caractère par caractère, à la recherche du p ième retour chariot (RETURN). Puis le pointeur de fichier est positionné à l'octet qui suit le p ième retour chariot. Si au cours de cette exploration, un octet dans lequel aucune information n'a été mémorisée, est rencontré, le message

END OF DATA

est affiché. Généralement cette situation se produit lorsque la p ième rubrique est située au-delà de la fin du fichier.

EXEC f (,Rp) (,Ss) (,Dd) (,Vv)

Exemple :

EXEC UTILITY

Cette commande est semblable à la commande RUN, sauf que le fichier f est en mode texte et qu'il contient des commandes BASIC et DOS, telles qu'elles sont tapées au clavier. La procédure cataloguée permet de contrôler automatiquement le déroulement des différentes opérations, comme vous le feriez vous-même à partir du clavier.

Il ne peut y avoir qu'une seule procédure cataloguée en fonction. Si la procédure cataloguée contient une commande EXEC, elle est fermée avant que la seconde procédure cataloguée ne prenne la main.

La commande CLOSE ne ferme pas le fichier de la procédure cataloguée. Celui-ci se ferme lui-même lorsqu'il a épuisé et exécuté toutes les commandes qu'il contenait. Si une procédure cataloguée contient une exécution de programme (RUN), elle attend patiemment que l'exécution soit terminée avant de reprendre la main.

ATTENTION : un programme exécuté par une procédure cataloguée ne peut pas comporter d'ordre INPUT. En effet, si c'était le cas, les caractères seraient acceptés à partir du fichier en EXEC, et non à partir du clavier. De plus, si ces caractères étaient une commande DOS, celle-ci s'exécuterait *avant* la suite du programme!

Autre danger : si vous interrompez un programme en BASIC étendu par CTRL-C alors qu'une procédure contrôle le système, les commandes non encore exécutées du fichier en EXEC seront *généralement ignorées.*

Si le paramètre R est utilisé, le pointeur de position dans le fichier est placé au début de la p ième rubrique du fichier, et la commande EXEC s'exécutera à partir de ce point.

De la même manière qu'avec la commande POSITION, le paramètre R devrait être considéré comme donnant une position relative dans le fichier. Cependant, contrairement à la commande POSITION, la commande EXEC compte toujours les rubriques à partir du début du fichier, si bien que p est toujours relatif à 0. Les autres paramètres fonctionnent comme habituellement. Si la valeur du paramètre R place le pointeur de fichier au-delà de la fin du fichier, le message

END OF DATA
sera affiché.

Les commandes pour les fichiers en mode texte à accès direct

OPEN f,Lj (,Ss) (,Dd) (,Vv)

Exemple :

OPEN SESAME,L2

Cette commande alloue une mémoire tampon de 595 octets pour le fichier en mode texte à accès direct f, et fixe la longueur d'enregistrement au nombre d'octets spécifié en j. La valeur de j peut varier de 1 à 32767. La valeur par défaut est 1.

OPEN est utilisé en liaison avec les commandes READ et WRITE, pour créer et consulter des fichiers en mode texte à accès direct. Remarquez que le paramètre L (Longueur) n'est pas optionnel : par définition, il est obligatoire de spécifier un paramètre L pour un fichier à accès direct.

Chaque fois qu'un fichier donné est utilisé, la commande OPEN doit comporter strictement la même valeur dans le paramètre L. En effet, le DOS utilise cette valeur pour calculer la position du premier octet de chaque enregistrement.

S'il n'y avait pas de fichier f, un fichier f serait créé.

CLOSE (f)

Exemple :

CLOSE BOOK

S'il s'agit d'un fichier en écriture, la commande CLOSE provoque l'écriture des caractères en mémoire tampon. La mémoire tampon du fichier f est désallouée. Si la commande CLOSE est utilisée sans nom de fichier, tous les fichiers ouverts sont fermés, sauf éventuellement le fichier en EXEC.

Si un programme est interrompu par un CTRL-C alors qu'un fichier en mode texte est ouvert, il est préférable de taper CLOSE pour éviter toute perte d'enregistrement.

ATTENTION : tout fichier qui a été ouvert (OPEN) doit être fermé (CLOSE), faute de quoi le contenu de la mémoire tampon peut être perdu (perte éventuelle des derniers enregistrements écrits).

WRITE f (,Rr) (,Bb)

Exemple :

WRITE ADDRESS.DATA,R3

Après cette commande, les ordres PRINT dirigent les caractères vers le fichier spécifié, au lieu de les diriger vers l'écran de visualisation.

La commande WRITE reste en fonctionnement jusqu'à
- une autre commande DOS;

- un ordre INPUT;
- une commande DOS nulle telle que PRINT CTRL-D.

La commande WRITE ne peut fonctionner qu'en mode différé (c'est-à-dire à partir d'un programme).

Le paramètre R (Record - enregistrement) a pour effet de faire débuter l'écriture au premier octet du r ième enregistrement contenant j octets, j étant donné avec le paramètre L dans la commande OPEN. La valeur par défaut de r est Ø. Si le paramètre B est utilisé, l'écriture débutera au b ième octet de r ième enregistrement du fichier.

Il convient de remarquer que lorsque la commande WRITE est en fonctionnement, tous les caractères qui seraient normalement affichés à l'écran sont écrits dans le fichier, y compris les éventuels messages d'erreur, points d'interrogation associés à un INPUT et autres caractères indésirables!

READ f (,Rr) (,Bb)

Exemple :

READ SESAME,R3,B3Ø

Après cette commande, les caractères sont acceptés par INPUT (et GET en BASIC étendu) à partir du fichier f à accès direct, au lieu du clavier.
L'ordre INPUT déclenche la *lecture d'une rubrique* de l'enregistrement concerné.

Une rubrique peut comporter théoriquement de 1 à 32767 octets, suivis d'un retour chariot (RETURN). Un enregistrement d'un fichier à accès direct ne peut comporter plus de j caractères, j étant la longueur donnée dans la commande OPEN.

Le paramètre R (Record - enregistrement) a pour effet de faire débuter la lecture au premier octet du r ième enregistrement, chaque enregistrement contenant j octets, j étant donné avec le paramètre L dans le commande OPEN. La valeur par défaut de r est Ø. Si le paramètre B est utilisé, la lecture débutera au b ième octet du r ième enregistrement du fichier. La commande READ reste en fonctionnement jusqu'à
- une quelconque commande du DOS;
- une commande du DOS nulle telle que PRINT CTRL-D.

Les commandes pour les fichiers en mode binaire

BSAVE f, Aa, Lj (,Ss) (,Dd) (,Vv)

Exemples :

BSAVE PICTURE, A16384, L8192
BSAVE PICTURE, A$4000, L$2000

La commande BSAVE crée un fichier nommé f, et y mémorise le contenu d'une portion de la mémoire centrale. La zone de mémoire est définie par son adresse de début a, et le nombre d'octets j à mémoriser.

Les exemples montrés mémorisent un graphique haute résolution en page 2. Du point de vue opérationnel, ils sont identiques : le second exemple utilise la notation hexadécimale au lieu de la notation décimale.

BLOAD f (,Aa) (,Ss) (,Dd) (,Vv)

Exemples :

BLOAD PICTURE, A8192
BLOAD PICTURE, A$2000

Si le paramètre a n'est pas indiqué, la commande BLOAD adopte comme adresse de début l'adresse de début utilisé lors de l'écriture (BSAVE). Si le paramètre a est indiqué, l'adresse de début en mémoire est l'adresse a.

Remarquez qu'un programme en langage machine ne sera pas utilisable si l'adresse est changée

Supposons que nous ayons mémorisé une page de graphique haute résolution sur disquette sous le nom de PICTURE. L'exemple ci-dessus chargerait PICTURE dans la première page de graphique haute résolution qui débute en adresse 8192 (décimale) ou $2000 (hexadécimale).

Note : si un BASIC étendu autre qu'en ROM était utilisé, il serait détruit en mémoire par l'exemple ci-dessus.

BRUN f (,Aa) (,Ss) (,Dd) (,Vv)

Exemple :

BRUN SUPER, A$C0A,V75

Cette commande charge (BLOAD) le programme en langage machine f en mémoire à partir de l'adresse a. Si le paramètre a est omis, la commande BRUN adopte comme adresse de début de chargement l'adresse de début utilisée lors de l'écriture (BSAVE). Une fois le programme chargé, son exécution débute grâce à un transfert de l'exécution par un code JMP (JUMP) à l'adresse a.

ANNEXE G
Sommaire des procédures du DOS

156 L'appel du DOS
156 L'initialisation d'une disquette
156 Comment faire si l'on passe accidentellement en mode Moniteur
156 L'utilisation du DOS à partir d'un programme
157 La création d'un système clé en main
157 La création et la consultation d'un fichier en mode texte à accès séquentiel
158 Ajouter des enregistrements à un fichier séquentiel
159 Le contrôle du système par une procédure cataloguée
159 La création et la consultation d'un fichier en accès direct
160 La copie d'un fichier en mode texte
160 Le recouvrement de programmes en BASIC étendu

L'appel du DOS

« s » représente le numéro de port (ou connecteur) dans lequel le contrôleur de la disquette est inséré.

Caractère d'appel	Langage	Pour appeler le DOS, taper
>	BASIC simple	PR#s ou IN#s
]	BASIC intégré	PR#s ou IN#s
★	Moniteur	s {CTRL} P

L'initialisation d'une disquette

Pour initialiser une disquette secondaire (qui est attachée à la taille mémoire du système sur lequel l'initialisation a été faite) :

1. Appeler le DOS.
2. Mettre une disquette vierge dans le lecteur.
3. Taper un programme de salutations, par exemple :

```
10 PRINT "32 K DISQUETTE SECONDAIRE INITIALISEE LE 5 MAI 1981"
20 END
```

4. En supposant que votre programme de salutations soit nommé « BONJOUR », taper la commande :

```
INIT BONJOUR
```

5. Lorsque la lampe témoin du lecteur de disquettes s'éteint complètement, retirer la disquette, et l'étiquetter.

Pour créer une disquette Maître, utilisez le programme MASTER CREATE (voir chapitre 5).

Comment faire si l'on passe accidentellement en mode Moniteur

Si le système est sous DOS et que la touche RESET a été frappée par erreur, taper 3D0G (3D zéro G) pour revenir à votre BASIC d'origine, avec votre programme intact.

L'utilisation du DOS à partir d'un programme

Les commandes du DOS peuvent être données à partir d'un programme. Elles doivent dans ce cas être précédées d'un ordre PRINT CTRL-D.

Il est préférable de définir au début du programme une variable contenant le caractère CTRL-D.
En BASIC étendu, on peut utiliser la fonction CHR$: D$ = CHR$(4), étant donné que la valeur ASCII (BASIC étendu) du CTRL-D est 4.
Dans les deux BASIC, on peut définir la variable du CTRL-D par D$ = "", un caractère CTRL-D (invisible à l'écran) étant tapé entre les guillemets.

Le programme suivant, écrit en BASIC étendu, affiche le catalogue lors de son exécution :

```
 5  REM PROGRAMME DE SALUTATIONS
10  D$ = CHR$(4) : REM CTRL-D
20  PRINT D$; "CATALOG"
```

Un ordre PRINT ne peut contenir qu'une seule commande du DOS. Il doit débuter avec le caractère CTRL-D et se terminer par la commande DOS. Le caractère CTRL-D doit être précédé d'un retour chariot (RETURN) émis à la fin de la plupart des ordres PRINT (attention aux ordres PRINT avec des points et virgules!).

Les commandes suivantes doivent être utilisées en mode différé :
OPEN APPEND READ WRITE POSITION

Il est préférable d'utiliser les commandes INIT et MAXFILES en commande directe.

Les autres commandes du DOS peuvent être utilisées indifféremment en mode direct ou en mode différé.

La création d'un système clé en main

Pour créer une disquette qui exécute un certain programme (dans cet exemple, COLOR DEMOS) à chaque fois que le DOS est appelé, procéder comme suit :

1. Initialiser une disquette vierge, avec par exemple le nom de BONJOUR comme programme de salutations.
2. Placer une disquette contenant le programme COLOR DEMOS dans le lecteur de disquettes, et taper :

   RUN COLOR DEMOS

   Revenir au BASIC par CTRL-D.
3. Placer la disquette initialisée dans le lecteur, puis taper :

   SAVE BONJOUR

pour remplacer l'ancien programme de salutations par COLOR DEMOS.

La création et la consultation d'un fichier en mode texte à accès séquentiel

Lors de la création d'un fichier en mode texte à accès séquentiel, une commande OPEN doit précéder la première commande WRITE. Sous l'effet d'une commande WRITE, tout ordre PRINT dirige les caractères à afficher vers la disquette. Un fichier doit être fermé lorsque les opérations le concernant sont terminées. Une commande WRITE reste en fonctionnement jusqu'à un ordre INPUT, une autre commande du DOS ou une commande DOS nulle telle que PRINT CTRL-D.

Le programme ci-dessous, écrit en BASIC étendu, crée un fichier séquentiel en mode texte nommé ÉCHANTILLON dont les treize premières rubriques contiennent trois messages suivis des nombres de 1 à 10 :

```
  5  REM ÉCRITURE DE ÉCHANTILLON
 10  D$ = CHR$ (4) : REM CTRL-D
 20  PRINT D$; "OPEN ECHANTILLON"
 30  PRINT D$; "DELETE ECHANTILLON"
 40  PRINT D$; "OPEN ECHANTILLON"
 50  PRINT D$; "WRITE ECHANTILLON"
 60  PRINT "PREMIER TEXTE" : PRINT "SECOND TEXTE"
 70  PRINT "TROISIÈME TEXTE"
 80  FOR J = 1 TO 10
 90  PRINT J : NEXT J
100  PRINT D$; "CLOSE ECHANTILLON"
```

Si l'on ouvre un fichier séquentiel existant, et que l'on donne une commande WRITE, les nouvelles rubriques viendront « écraser » celles qui étaient préalablement écrites au début du fichier.

Le programme ci-dessous, écrit en BASIC étendu, consulte le fichier ÉCHANTILLON préalablement écrit, rubrique par rubrique. Si l'on désire voir les caractères lus, on peut taper MON I.

```
  5  REM LECTURE DU FICHIER ÉCHANTILLON
 10  D$ = CHR$ (4) : CHR$ (4) EST LE CTRL-D
 20  PRINT D$; "OPEN ECHANTILLON"
 30  PRINT D$; "READ ECHANTILLON"
 40  INPUT A$, B$, C$
 50  FOR I = 1 TO 10
 60  INPUT W
 70  NEXT I
 80  PRINT D$; "CLOSE ECHANTILLON"
```

La commande OPEN doit précéder la première commande READ d'un programme. Sous l'effet d'une commande READ, tout ordre INPUT (ou GET en BASIC étendu) accepte les caractères à partir de la disquette au lieu du clavier.

Un fichier doit être fermé lorsque les opérations le concernant sont terminées.

Une commande READ reste en fonctionnement jusqu'au prochain ordre PRINT CTRL-D, qu'il soit suivi ou non d'une commande du DOS.

Ajouter des enregistrements à un fichier séquentiel

Le programme ci-dessous, écrit en BASIC étendu, permet de compléter le fichier ÉCHANTILLON avec deux rubriques : TEXTE 4 et TEXTE 5.

```
  5  REM APPEND ECHANTILLON
 10  D$ = CHR$ (4) : REM CTRL-D
 20  PRINT D$; "APPEND ECHANTILLON"
 30  PRINT D$; "WRITE ECHANTILLON"
 40  PRINT "TEXTE 4"
 50  PRINT "TEXTE 5"
 60  PRINT D$; "CLOSE ECHANTILLON"
```

Le contrôle du système par une procédure cataloguée

Lorsque le programme en BASIC étendu ci-dessous est exécuté, il crée un fichier en mode texte nommé A FAIRE, et contenant les commandes LIST 20, 50

```
RUN BONJOUR
CATALOG
```

```
 5  REM CRÉATION A FAIRE
10  D$ = CHR$ (4) : REM CTRL-D
20  PRINT D$; "OPEN A FAIRE"
30  PRINT D$; "WRITE A FAIRE"
40  PRINT "LIST 20, 50"
50  PRINT "RUN BONJOUR"
60  PRINT "CATALOG"
70  PRINT D$; "CLOSE A FAIRE"
```

Une fois que le fichier A FAIRE est créé, la commande

```
EXEC A FAIRE
```

permettra au fichier A FAIRE de contrôler le système, et d'exécuter une à une les commandes qu'il contient comme si elles avaient été tapées au clavier.

La création et la consultation d'un fichier en accès direct

Le prochain programme écrit en BASIC étendu crée un fichier en mode texte à accès direct nommé RA-FILE, dont les enregistrements font 30 octets de long.
Puis les enregistrements de numéro d'ordre 12 à 15 sont écrits avec le libellé « NOM ADRESSE » suivi du numéro d'ordre de l'enregistrement. Enfin, en lignes 70 et 80, l'enregistrement d'ordre 13 est modifié pour y mémoriser le message « DOS »

```
  5  REM CRÉATION DE RA-FILE
 10  D$   D$ = CHR$ (4) : REM CTRL-D
 20  PRINT D$; "OPEN RA-FILE"
 30  PRINT D$; "DELETE RA-FILE"
 40  PRINT D$; "OPEN RA-FILE, L30"
 50  FOR I = 12 TO 15
 60  PRINT D$; "WRITE RA-FILE, R"; I
 70  PRINT "NOM ADRESSE"; I
 80  NEXT I
 90  PRINT D$; "WRITE RA-FILE, R13"
100  PRINT "DOS"
110  PRINT D$; "CLOSE RA-FILE"
```

L'exemple ci-dessous, écrit en BASIC étendu, lit les enregistrements 12 à 15 du fichier à accès direct RA-FILE. Remarquez qu'il faut donner le numéro d'ordre de chaque enregistrement, en ligne 40. La ligne 60 examine les trois premiers caractères de chaque enregistrement. Si ces trois premiers caractères sont "DOS", le message « l'enregistrement r a été modifié » est affiché, et la recherche continue.

```
 5  REM CONSULTATION DE RA-FILE
10  D$ = CHR$ (4) : REM CTRL-D
20  PRINT D$; "OPEN RA-FILE, L30"
30  FOR J = 12 TO 15
40  PRINT D$; "READ RA-FILE, R"; J
50  INPUT A$
60  IF LEFT$(A$,3) = "DOS" THEN PRINT "L'ENREGISTREMENT"; J; "A ÉTÉ
MODIFIÉ"
70  NEXT J
80  PRINT D$; "CLOSE RA-FILE"
```

La copie d'un fichier en mode texte

Copier un programme en mode BASIC ou en mode binaire (ou un fichier en mode binaire) d'une disquette à une autre est très facile : il suffit de faire LOAD (ou BLOAD), puis SAVE (ou BSAVE) sur la nouvelle disquette. Rappelez-vous d'indiquer la longueur et l'adresse de départ lorsque vous utilisez la commande BSAVE.

Le programme COPY permet de copier la totalité d'une disquette sur une autre. Si l'un des fichiers ou des programmes est protégé contre les copies, le message

```
I/O ERROR
STOPPED AT 320
```

sera affiché.

Pour copier un fichier en mode texte, ou un fichier quelconque, à condition qu'il ne soit pas protégé, vous pouvez utiliser le programme FID. Le mode d'utilisation du programme FID est développé dans l'annexe J.

Le recouvrement de programmes en BASIC étendu

Le recouvrement consiste à exécuter une série de programmes en BASIC étendu sans perdre les valeurs des variables et des tables.

Supposons que vous désiriez enchaîner un programme nommé PARTIE 2 à un programme nommé PARTIE 1. Assurez-vous que le programme CHAIN est sur la même disquette que les deux programmes (s'il n'y est pas, enregistrez-le en utilisant le programme FID). Il suffit d'ajouter ces deux lignes comme dernières lignes du programme PARTIE 1 :

```
PRINT CHR$ (4); "BLOAD CHAIN, A520"
CALL 520 "PARTIE 2"
```

Il est essentiel qu'il n'y ait aucun espace entre le 0 de 520 et le guillemet dans la commande CALL.

ANNEXE H
Comment passer au DOS 16 secteurs

162 Comment installer les nouvelles PROMS
163 Système à plus de un lecteur de disquettes
164 L'utilisation du DOS avec la carte langage
164 La réalisation d'un système clé en main avec la carte langage

Si vous avez déjà un DOS (Disk Operating System), et que celui-ci fonctionne avec des disques formatés à 13 secteurs (la version 3.2.1. du DOS, ou une version antérieure), il sera nécessaire de changer deux PROMS sur la carte d'interface de votre lecteur de disquettes, avant de pouvoir passer le système à un formatage à 16 secteurs. Toute version antérieure au 3.3 nécessitera cette mise à jour.

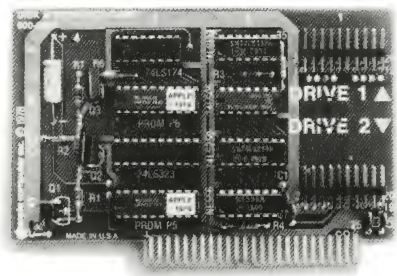
Comment installer les nouvelles PROMS

Les nouvelles PROMs remplacent deux PROMs sur chaque carte d'interface de lecteurs de disquettes.

A l'intérieur du boîtier de votre micro-ordinateur, sur la partie la plus éloignée du clavier, se trouvent les connecteurs d'interfaces. Dans l'un ou dans plusieurs de ces connecteurs, vous trouverez une carte portant la mention "DISK II INTERFACE CARD" sur le bord supérieur. Assurez-vous que l'appareil est hors tension. Retirez chacune de ces cartes doucement, par un mouvement de va et vient pour faciliter la sortie de la carte de la fente du connecteur. Retirez les câbles de connexion aux lecteurs de disquettes afin de pouvoir opérer plus aisément.

Tenez la carte d'interface dans une main, côté composants vers vous, la partie rayée étant vers le bas. Vous verrez quatre rangées de deux composants sur la carte. Repérez le composant de gauche, dans la seconde rangée à partir du haut. Il devrait être référencé "P6", ou "341-0010-XX" (il est possible que la référence soit recouverte d'une étiquette portant mention du Copyright), et l'indication "PROM P6" doit être portée immédiatement sur la carte, sous ce composant. C'est celui que vous remplacerez par la nouvelle PROM P6A.

Repérez maintenant le composant de gauche sur la rangée du bas. Il devrait être référencé "P5" ou "341-0009-XX" (peut-être sous étiquette adhésive), et l'indication "PROM P5" doit être portée immédiatement sous ce composant. C'est celui que vous remplacerez par la nouvelle PROM P5A.



Examinez maintenant les nouvelles PROMs. La nouvelle PROM P6A doit être marquée "P6A" ou "341-0028-XX", ou les deux. La nouvelle PROM P5A doit être marquée "P5" ou "341-0027-XX" ou les deux. Vous devriez pouvoir trouver l'une de ces références sans déplacer les étiquettes adhésives. Si vous ne la trouvez pas, voyez votre distributeur.

Maintenant que vous savez où se situent les différents éléments, vous êtes prêt à remplacer les anciennes PROMS par les nouvelles. Posez la carte d'interface sur une table, et maintenez-la fermement. En utilisant la pince à composants, retirez doucement la PROM P6 de son logement. Il n'est pas impossible qu'un

transistor marqué Q3 gêne pour retirer la PROM P6. Dans ce cas, redressez-le doucement afin qu'il ne gêne plus. Puis retirez la PROM P6, en la faisant très légèrement basculer de part et d'autre, pour faciliter le mouvement.

Tenez la PROM P6A, référencée "P6A" ou "341-0028-XX" dans le même sens que les composants de la carte. La grande découpe doit se trouver sur le côté gauche lorsque vous êtes face à la carte (il est possible qu'il y ait une petite découpe circulaire à l'autre extrémité du composant. Si vous n'êtes pas certain du sens dans lequel il convient de mettre la PROM, demandez à votre distributeur). Assurez-vous que chaque pointe est face au trou correspondant, et insérez doucement la PROM P6A dans le logement P6. Si les pointes sont trop écartées pour pouvoir entrer aisément entre les trous, il faut les remettre en position verticale, en exerçant une légère pression sur l'ensemble des pointes, sur la surface d'une table, par exemple.

Retirez maintenant la PROM P5, et remplacez-la par la PROM P5A, référencée "P5A" ou "341-0027-xx", exactement de la même manière.

Si vous avez plus de deux lecteurs de disquettes DISK II, il faut disposer d'une carte d'interface pour chaque paire de lecteurs. Pour chaque carte d'interface, il faudra changer les deux PROMS. Procédez de la même manière que pour la première.

Système à plus de un lecteur de disquettes

Connectez chaque lecteur DISK II à sa carte d'interface, et remettez les cartes d'interface dans les connecteurs du micro-ordinateur, dans l'ordre donné ci-dessous. Le premier lecteur doit être mis dans le lecteur 1 du port (connecteur) 6 (slot 6, drive 1), et ainsi de suite jusqu'à ce que tous les lecteurs soient reliés. En commençant par le port 6, lecteur 1, puis lecteur 2, pour chaque port, vous obtiendrez la meilleure performance possible, et accroîtrez la fiabilité de votre système : les programmes ne tenteront pas d'utiliser des lecteurs qui ne sont pas connectés, et n'ignoreront pas ceux qui le sont.

POSITION DES LECTEURS DE DISQUETTES DIKS II

Ordre d'installation	Nº de port	Nº de lecteur
1º lecteur	6	1
2º lecteur	6	2
3º lecteur	5	1
4º lecteur	5	2
5º lecteur	4	1
6º lecteur	4	2

Collez les étiquettes DRIVE 1 (lecteur 1) et DRIVE 2 (lecteur 2) sur les lecteurs appropriés. Vérifiez que tout est bien en place. La PROM P5 de chaque carte d'interface doit avoir été remplacée par la PROM P5A (341-0027-XX). La PROM P6 de chaque carte d'interface doit avoir été remplacée par la PROM P6A (341-0028-XX). Un câble ruban doit être connecté à la partie supérieur (lecteur 1) de chaque carte d'interface. Il doit y avoir une carte d'interface dans le port 6.

Si tout a l'air bien installé, replacez le couvercle sur votre micro-ordinateur. Vous allez pouvoir passer vos disquettes à un formatage de 16 secteurs par piste, en utilisant le programme MUFFIN.

L'utilisation du DOS avec la carte langage

Vous pouvez utiliser le DOS avec le BASIC étendu APPLESOFT ou avec le BASIC entier (Integer) alors que la Carte Langage est installée. Il n'est pas nécessaire d'utiliser la disquette BASICS. Pour passer au "DOS 16 secteurs" après avoir utilisé un quelconque langage avec la Carte Langage fonctionnant sous le système d'exploitation du PASCAL (PASCAL Operating System), il suffit de procéder comme suit :

1. Assurez-vous que vous avez sauvegardé le programme qui se trouve en mémoire centrale si vous désirez le conserver. En effet, il sera effacé lors de l'appel du DOS.

2. Passez en mode Commande du système PASCAL (la ligne qui débute par "COMMAND : " est affichée).

3. Insérez la disquette Maître du DOS dans le lecteur 1 (celui qui est connecté au port de numéro le plus élevé).

4. Tapez H (pour Halt). Ceci aura pour effet d'appeler le DOS à partir de la disquette Maître.

Votre système chargera soit le BASIC étendu APPLESOFT, soit le BASIC entier — celui des deux qui n'est pas déjà disponible dans la ROM d'origine — dans la zone de travail disponible sur la Carte Langage. Si vous disposez d'un APPLE II standard, le système Maître chargera l'APPLESOFT dans la zone de travail de la Carte Langage. Si vous disposez d'un APPLE II PLUS, c'est le BASIC entier qui sera chargé. Dans les deux cas, un message sera affiché, vous indiquant quel est le BASIC dont le chargement est en cours. Toutes les commandes du DOS sont maintenant accessibles à la fois en BASIC étendu, et en BASIC entier.

Pour maîtriser la procédure permettant d'utiliser le DOS après un langage qui ne fait pas appel au Système d'Exploitation du Pascal, référez-vous au Manuel du langage de programmation concerné.

Avec la Carte Langage installée, vous pouvez appeler le DOS sans manipulation, simplement en mettant le micro-ordinateur sous tension.

Pour retourner à un autre langage après le DOS, mettre la disquette de démarrage de ce langage dans le lecteur 1. Tapez PR#6, puis frappez la touche RETURN.

La réalisation d'un système clé en main avec la carte langage

Lorsque vous insérez une disquette DOS 3.3 dans le lecteur 1 et que vous mettez le micro-ordinateur sous tension, le programme de salutations de cette disquette s'exécute si le BASIC dans lequel il est écrit est disponible en ROM. Plus précisement, un APPLE II standard exécutera un programme de salutations en BASIC entier; un APPLE II PLUS exécutera un programme de salutations en BASIC étendu; un APPLE II standard avec la carte APPLESOFT, ou un APPLE II PLUS avec la carte BASIC entier peuvent exécuter un programme de salutations dans un BASIC ou dans un autre. Si le BASIC concerné n'est pas disponible en ROM, le message

LANGUAGE NOT AVAILABLE

sera affiché, suivi, à la ligne, du caractère d'appel du BASIC disponible en ROM.

Si vous disposez de la Carte Langage, la situation se complique quelque peu. Vous pouvez sans problème exécuter les programmes qui sont écrits dans le BASIC de la ROM d'origine. Mais avant que vous ne puissiez utiliser l'autre BASIC, il sera nécessaire de le charger sur la Carte Langage. La façon la plus simple de procéder est d'appeler le DOS 3.3 à partir de la disquette Maître qui comporte deux programmes nommés HELLO et APPLESOFT. Ces deux programmes se chargeront de faire en sorte que les deux BASICS soient disponibles en ROM lorsque le DOS sera appelé. Le programme HELLO, écrit en APPLESOFT pour l'APPLE II PLUS, charge le BASIC entier. Le programme APPLESOFT, écrit en BASIC entier pour l'APPLE II standard, charge le BASIC étendu APPLESOFT. Votre micro-ordinateur exécutera le programme qui convient, et se comportera ensuite comme s'il disposait des deux BASICS en ROM. Vous pouvez maintenant insérer une disquette DOS 3.3 et exécuter le programme que vous désirez.

Si vous désirez constituer une disquette qui démarrera automatiquement sur n'importe quel système et qui réalisera le chargement de la Carte Langage si nécessaire, il faut y mémoriser les programmes appropriés. Voici comment :

1. Appelez le DOS 3.3 à partir d'une disquette Maître.

2. Chargez le programme HELLO en mémoire, insérez une disquette vierge et tapez INIT HELLO.

3. Lorsque l'initialisation est terminée, remettez la disquette Maître dans le lecteur et tapez BRUN MASTER CREATE. Ce programme vous demandera le nom du programme de salutations. Tapez HELLO. Insérez la disquette initialisée lorsque MASTER CREATE vous le demande.

4. En utilisant le programme FID, copiez sur la disquette nouvellement initialisée les programmes :

APPLESOFT
INTBASIC
FPBASIC

5. Vous pouvez charger vos programmes et vos fichiers sur la disquette. Assurez-vous que votre catalogue comporte bien au moins :

HELLO
APPLESOFT
INTBASIC
FPBASIC

Lorsque vos démarrerez avec cette disquette sur une machine comportant la Carte Langage, le BASIC qui n'est pas dans la ROM d'origine sera chargé sur la Carte Langage, afin que les programmes écrits dans les deux langages BASIC puissent être exécutés.

Si vous voulez disposer d'un système clé en main, vous pouvez modifier les programmes HELLO et APPLESOFT, afin que chacun lance l'exécution de votre programme d'appel. Nous le nommerons CLE-EN-MAIN.

Avant d'afficher ces programmes, il sera plus commode d'exécuter au préalable le programme LOADAPA par RUN LOADAPA, puis d'utiliser la commande &S afin de voir les caractères CTRL-D qui sont insérés dans les commandes du DOS.

Démarrez avec la disquette que vous avez constituée en effectuant les opérations de 1 à 5, puis procédez comme suit :

6. UNLOCK HELLO

7. LOAD HELLO

8. LIST

examinez les dernières lignes :

```
240  END
250  REM
260  REM — NO CARD OR CAN'T RELOAD
270  REM
280  IF PEEK (768) =0 THEN END
290  PRINT : PRINT "...LANGUAGE CARD CANNOT BE RELOADED" :
     PRINT "UNTIL THE SYSTEM IS REBOOTED..."
300  END
```

A l'écran, ces lignes apparaissent sur 40 colonnes. Elles sont disposées autrement ici, pour plus de clareté.

9. Modifiez ces lignes pour obtenir :

```
* 240  GOTO 300
  250  REM
  260  REM — NO CARD OR CAN'T RELOAD
  270  REM
* 280  IF PEEK(768) = 0 THEN 300
  290  PRINT : PRINT "...LANGUAGE CARD CANNOT BE RELOADED" :
       PRINT "UNTIL THE SYSTEM IS REBOOTED..."
* 300  PRINT "crtl-d RUN CLE-EN-MAIN"
```

Les lignes modifiées ou ajoutées sont repérées par des astérisques. Le CTRL-D en ligne 300 représente un caractère CTRL- qui n'apparaîtra pas, ni à l'écran, ni sur une liste à l'imprimante. Si vous n'aimez pas travailler avec des CTRL-D invisibles, vous pouvez déclarer une variable alphanumérique D$ contenant ce caractère, ce qui donnerait pour la ligne 310 (en APPLESOFT) :

```
* 310  D$ = CHR$ (4) : PRINT D$; "RUN CLE-EN-MAIN"
```

Si vous adoptez cette méthode, assurez-vous que vous avez défini le D$ avant de l'utiliser.

Il est possible aussi que vous souhaitiez modifier ou supprimer certains des ordres PRINT de ce programme. Dans ce cas, assurez-vous que vous ne supprimez pas les ordres PRINT qui concernent les commandes du DOS (ils comportent des caractères CTRL-D).

10. SAVE HELLO

11. LOCK HELLO

12. UNLOCK APPLESOFT - LOAD APPLESOFT - LIST.

Examinez les dernières lignes :

```
250  GOTO 310
260  REM
270  REM — NO CARD OR CAN'T RELOAD
280  REM
290  IF PEEK(768) =0 THEN 310
300  PRINT "...LANGUAGE CARD CANNOT BE RELOADED" :
     PRINT "UNTIL THE SYSTEM IS REBOOTED"
310  PRINT "ctrl-dINT"
```

13. Modifiez ces lignes pour obtenir :

```
  250   GOTO 310
  260   REM
  270   REM — NO CARD OR CAN'T RELOAD
  280   REM
* 290   IF PEEK(768) = 0 THEN 310
  300   PRINT ...LANGUAGE CARD CANNOT BE RELOADED" :
        PRINT "UNTIL THE SYSTEM IS REBOOTED"
* 310 PRINT "ctrl-dRUN CLE-fEN-MAIN"
```

La ligne 310 peut aussi être écrite sous la forme :

```
* 310 D$ ="ctrl-d" : PRINT D$; "RUN CLE-EN-MAIN"
```

afin de souligner la présence du caractère CTRL-D.

14. SAVE APPLESOFT

15. LOCK APPLESOFT

16. Renommez le programme d'appel de votre système clé en main, pour l'appeler CLE-EN-MAIN, s'il ne porte pas déjà ce nom, en utilisant la commande RENAME.
A titre d'exemple, vous trouverez ci-dessous un programme du type clé en main, écrit en APPLESOFT :

```
100   TEXT : HOME
110   VTAB3 : PRINT "DOS TOOL KIT"; TAB(30); "30NOV 1979"
120   VTAB5 : PRINT "(C) COPYRIGHT 1979, APPLE COMPUTER INC."
130   FOR I = 1 TO 1500 : NEXT I
140   PRINT "ctrl-dRUN RIBBIT"
```

17. Assurez-vous que votre disquette comporte les programmes suivants :

HELLO
APPLESOFT
INTBASIC
FPBASIC
CLE-EN-MAIN

18. Faites un essai de votre système clé en main, en introduisant la disquette dans le lecteur 1. Eteignez, puis rallumez votre système. Le DOS 3.3 doit être appelé automatiquement; puis le BASIC entier doit se charger sur la Carte Langage, si nécessaire, et enfin, le programme CLE-EN-MAIN s'exécute.

ANNEXE I
L'utilisation de la disquette BASICS

170 Le démarrage des disquettes avec « BASICS »

La disquette BASICS qui accompagne la disquette Maître est nécessaire pour utiliser des disquettes formatées en 13 secteurs, sans les modifier, sur votre système qui prévoit des disquettes formatées sur 16 secteurs. Les versions du DOS antérieures au 3.3 formatent les disquettes en 13 secteurs par piste.

APPLE est disponible en deux « variétés » : APPLE II et APPLE II Plus. L'APPLE II comporte à l'origine le langage BASIC entier en ROM. L'APPLE II Plus comporte à l'origine le BASIC étendu en ROM. Le langage en ROM d'origine est celui dans lequel se mettra l'ordinateur lorsqu'il est mis sous tension, s'il ne comporte pas de lecteur de disquettes ou de carte d'interface spéciale.

Le démarrage des disquettes avec « BASICS »

Pour utiliser soit le BASIC entier, soit l'APPLESOFT, insérez la disquette « BASICS » dans le lecteur, et mettez l'appareil sous tension. Après environ cinq secondes, le message suivant s'affiche à l'écran :

INSERT YOUR 13 SECTOR DISKETTE
AND PRESS RETURN
(Insérez votre disquette 13 secteurs
et frappez RETURN)

Insérez une disquette BASIC quelconque formatée à 13 secteurs, et frappez la touche « RETURN ». L'APPLE se comportera comme écrit dans ce manuel, sauf qu'une opération est ajoutée à la procédure de démarrage : lorsque vous mettez l'appareil sous tension, ou lorsque vous tapez PR#6, IN#6, C600G ou 6 CTRL-P, la disquette « BASICS » doit être dans le lecteur, chaque fois que vous désirez utiliser une disquette formatée 13 secteurs.

Lorsque le message apparaît à l'écran, insérez la disquette BASIC 13 secteurs que vous désirez utiliser. Si vos utilisez l'un ou l'autre des langages BASIC, et que vous tapiez PR#6 ou IN#6, il faudra d'abord insérer la disquette « BASICS » dans le lecteur, puis y remettre votre disquette à nouveau.

Une fois que le DOS est appelé, vous pouvez passer normalement d'un DOS à l'autre, en utilisant les commandes INT et FP.

Comme d'habitude, vous pouvez voir dans quel langage BASIC vous travaillez par la présence du caractère d'appel : > pour le BASIC entier,] pour le BASIC étendu APPLESOFT.

Une disquette prévue pour un système clé en main avec un démarrage automatique peut être exécuté comme précédemment, après que la disquette « BASICS » ait été utilisée comme nous venons de le voir.

Vous pouvez appeler de nouveau le DOS sans éteindre la machine. Il suffit d'insérer la disquette « BASICS », de taper PR#6 ou IN#6, puis RETURN. A l'apparition du message à l'écran, insérer la disquette que vous désirez démarrer.

ANNEXE J
L'utilisation du programme FID

172 Le démarrage
172 Les noms de fichier et les caractères génériques
173 Les commandes
177 Que faire en cas d'erreur

FID est l'abréviation de File Developer (créateur de fichier). Ce programme étend les possibilités du DOS à 16 secteurs de deux manières :

- il permet l'utilisation aisée de CATALOG (catalogue), COPY (copier), DELETE (supprimer), LOCK (verrouiller), UNLOCK (déverrouiller) de tous les fichiers sous DOS;
- il permet la copie d'une disquette à une autre, en n'utilisant qu'un seul lecteur de disquettes.

FID peut fonctionner avec n'importe quel APPLE II comportant l'APPLESOFT, 32 K de mémoire RAM ou plus, et un lecteur de disquettes, ou plus. La disquette réceptrice (celle sur laquelle FID copiera) doit être préalablement initialisée.

Le démarrage

Pour utiliser le programme FID, appelez le DOS à partir d'une disquette Maître, puis tapez :
BRUN FID

Vous verrez apparaître une liste de commandes à l'écran. Chacune peut être utilisée simplement en tapant le numéro entre <>.

Les noms de fichiers et les caractères génériques

Certaines commandes nécessiteront que vous fournissiez un nom de fichier. Le nom de fichier est tapé exactement comme pour les commandes du DOS, sauf que vous pouvez remplacer une partie du nom du fichier par le caractère générique (=). Par exemple, si vous tapiez le nom de fichier

FR=ED

le programme rechercherait tous les noms de fichiers débutant par FR et se terminant par ED. Si vous tapiez

=N=

tos les fichiers dont le nom contient la lettre N seraient sélectionnés. Si vous tapiez

=

tous les fichiers de la disquette seraient retenus (sur laquelle disquette? vos demandez-vous. Vous trouverez les explications ci-contre).

Si vous avez utilisé un caractère générique en donnant un nom de fichier, l'ordinateur vous demandera

DO YOU WANT PROMPTING?
(Désirez-vous être consulté?)

Si vous répondez Y pour Yes (oui), le système vous demandera de vérifier chaque nom de fichier, avant qu'il n'intervienne. Dans ce cas, si vous désirez traiter le fichier, répondez Y lorsque son nom apparaîtra à l'écran. Si vous ne désirez pas le traiter, tapez N.

Dans les deux cas, le fichier suivant vous sera présenté. Si vous ne souhaitez pas traiter ce fichier, ni aucun de deux qui suivent, il suffit de taper Q et les autres fichiers ne vous seront pas présentés.

Si vous répondez N à la question "DO YOU WANT PROMPTING?", tous les fichiers correspondant à votre demande seront retenus.

Les commandes

La plupart des commandes de FID (CATALOG, SPACE, DELETE, LOCK, UNLOCK et VERIFY) fonctionnent sur un seul lecteur : le dernier lecteur utilisé, à moins d'une restauration du lecteur par défaut.

La commande COPY peut utiliser un ou deux lecteurs, à votre choix. La commande QUIT n'utilise aucun lecteur. Elle a simplement pour effet de vous faire quitter le programme FID, et de vous ramener au DOS.

Au début de l'exécution du programme, il n'y a pas de numéro de lecteur par défaut. Si la première commande que vous utilisez est une commande qui n'utilise qu'un seul lecteur de disquettes, CATALOG, SPACE, DELETE, LOCK, et UNLOCK, le système vous demandera un numéro de port et un numéro de lecteur. Ils deviendront les numéros par défaut jusqu'à ce qu'ils soient modifiés. Si votre première commande est la commande COPY, le système vous demandera deux numéros de port et de lecteur, l'un pour la disquette originale (émettrice), l'autre pour la disquette copie (réceptrice).

Chaque fois que vous passerez de la commande COPY à une commande à un seul lecteur, et inversement, le système vous demandera d'indiquer les caractéristiques de un ou de deux lecteurs. A l'aide de la commande RESET vous pouvez annuler les numéros de port et de lecteur par défaut. FID vous demandera de lui en indiquer de nouveaux, si nécessaire.

REMARQUE : Assurez-vous que vous indiquez des numéros de port et de lecteur qui sont bien connectés. Si l'ordinateur essaie de lire ou d'écrire à partir d'un lecteur inexistant, l'exécution s'arrête.

CATALOG

Le fait de taper 2 fait apparaître le catalogue de la disquette insérée dans le lecteur par défaut. Si vous n'avez pas encore donné le numéro de port et le numéro de lecteur par défaut, le système vous les demandera. Une fois que vous les avez donnés, toutes les commandes se référeront à ce lecteur, sauf si le port/lecteur par défaut est restauré.

La restauration (RESET) du port et du lecteur

La commande RESET vous permet de changer les numéros de port et de lecteur par défaut. La prochaine fois que vous donnerez une commande qui nécessite l'utilisation d'un numéro de port et de lecteur, le système vous le demandera. Cette commande est commode si vous désirez consulter successivement le catalogue de deux disquettes.

REMARQUE : La commande de restauration du numéro de port et du numéro de lecteur ne doit pas être confondue avec la touche RESET. Ils sont représentés en caractères différents dans ce Manuel, afin que le lecteur les distingue bien.

LOCK FILE (Verrouiller un fichier)

Pour verrouiller un fichier, tapez 5, et donnez le nom de fichier lorlque le système vous le demande. Si vous tapez un nom de fichier invalide, le système vous redemandera un nom, jusqu'à ce que vous en indiquiez un valide.

Puis le système vous engagera à insérer la disquette contenant le fichier dans le lecteur, et vous proposera soit d'interrompre l'opération par ESC (Escape), soit de poursuivre.

Si vous poursuivez, et que le fichier dont vous avez donné le nom est bien sur la disquette, il sera verrouillé. Vous pourrez le constater par la commande CATALOG : le nom du fichier verrouillé apparaîtra précédé d'un astérisque, *. Si le fichier n'est pas sur la disquette, le message

NO FILES SELECTED

sera affiché.

Vous pouvez recommencer, en frappant une touche quelconque sauf RESET, SHIFT, CTRL ou ESC

UNLOCK FILE (Déverrouiller un fichier)

Pour déverrouiller un fichier, tapez 4 et donnez le nom de fichier lorsque le système vous le demande. Si vous tapez un nom invalide, le système vous redemandera un nom jusqu'à ce que vous lui en indiquiez un valide.

Puis le système vous engagera à insérer la disquette contenant le fichier dans le lecteur, et vous proposera soit d'interrompre l'opération par ESC (Escape), soit de poursuivre.

Si vous poursuivez, et que le fichier dont vous avez donné le nom soit bien sur la disquette, il sera déverouillé. Vous pourrez le constater par la commande CATALOG : le nom du fichier déverouillé apparaîtra sans être précédé d'un astérisque, *. Si le fichier n'est pas sur la disquette, le message

NO FILES SELECTED

sera affiché.

Vous pourrez recommencer en frappant une touche quelconque, sauf RESET, SHIFT, CTRL, ou ESC.

VERIFY (Vérifier)

La commande VERIFY est la même que la commande VERIFY du DOS; elle concerne toutes les natures de fichier. Pour l'utiliser, tapez 8 et conformez-vous aux instructions données. Lorsqu'un fichier a été vérifié, son nom s'affiche à l'écran, suivi du mot

DONE (fait)

Si un fichier ne peut être lu, il s'agit d'un fichier invalide, et c'est le message I/O ERROR (erreur d'entrée/sortie) qui apparaîtra. Dans ce cas, le système vous engagera à frapper une touche quelconque, ce qui ramènera l'exécution au menu principal.

COPY FILE (Copie de fichier)

FID peut réaliser des copies de fichier, en utilisant soit un seul lecteur de disquettes, soit deux lecteurs de disquettes.

Pour copier des fichiers, tapez 1. Vous devez indiquer ensuite les numéros de port et de lecteur de la disquette originale (émettrice) et les numéros de port et de lecteur de la disquette copie (réceptrice).

La copie utilisant plus de un lecteur de disquettes.
Si les numéros de port et de lecteur indiqués sont différents, un nom de fichier vous sera demandé. Il peut contenir des caractères génériques. Vous pouvez, par exemple, copier la totalité d'une disquette en

répondant simplement =. Lorsque vous aurez indiqué le numéro de fichier, le système vous engagera à insérer les disquettes concernées dans les lecteurs et à frapper

soit la touche ESC (Escape), pour interrompre;
soit une touche quelconque, sauf RESET, SHIFT, CTRL ou ESC, pour poursuivre.

Si les fichiers que vous avez demandés sont sur la disquette d'origine (émettrice), et s'il y a assez de place sur la disquette copie (réceptrice), vous serez informé que la copie a été réalisée. Sinon, vous obtiendrez un message d'erreur du DOS.

Si la disquette copie (réceptrice) contient déjà un fichier portant le nom indiqué, le message

FILE
ALREADY EXISTS
TYPE IN NEW FILE NAME FOR THE COPY OR
<RETURN> TO REPLACE PRESENT FILE OR
<CTRL-C> <RETURN> TO CANCEL COPY
:

(Le fichier existe déjà. Donnez un autre nom de fichier pour la copie ou

<RETURN> pour remplacer l'ancien fichier par la copie
<CTRL-C> pour annuler la copie).

Si vous donnez un nouveau nom de fichier, le fichier sera recopié, et mémorisé sur la disquette copie (réceptrice) sous le nouveau nom. Si vous frappez la touche RETURN, l'ancien fichier sera remplacé par la copie, à condition que l'ancien fichier ne soit pas verrouillé. S'il est verrouillé, le message

FILE LOCKED
DO YOU WISH TO REPLACE IT ANYWAY?
(Fichier verrouillé
désirez-vous vraiment le perdre en le remplaçant?)

Si vous tapez Y (yes - oui), le fichier sera «écrasé».
Si vous tapez N (no - non), le message

FILE ALREADY EXISTS
TYPE IN NEW FILE NAME FOR THE COPY OR
<RETURN> TO REPLACE PRESENT FILE OR
<CTRL-C> <RETURN> TO CANCEL COPY
:

Si vous tapez CTRL-C puis RETURN, le fichier ne sera pas copié. Si le programme ne trouve pas les fichiers à copier, le message

NO FILES SELECTED
(Aucun fichier retenu)

sera affiché.

Si le fichier a été trouvé, le programme affiche le message

DONE
(fait)

Puis le système vous engagera à frapper une touche quelconque :

PRESS ANY KEY TO CONTINUE

Le fait de frapper une touche quelconque ramènera l'exécution au menu.

La copie utilisant un seul lecteur de disquettes.
Si les numéros de port et de lecteur indiqués sont identiques pour la disquette émettrice et pour la disquette réceptrice, le programme vous indiquera à quel moment il convient d'insérer l'original (émettrice) et la copie (réceptrice). Ainsi, on peut réaliser des copies de disquettes même si l'on ne dispose que d'un seul lecteur.

DELETE FILE (Suppression de fichier)

Cette commande supprime un fichier, ou un ensemble de fichiers, de la disquette par défaut.

Pour l'utiliser, tapez 6. Le programme vous demandera

FILE NAME ?
(Quel est le nom du fichier ?)

Lorsque vous donnerez le nom du fichier, le programme tentera de supprimer le fichier portant ce nom. Si tout se passe bien, le message

<DONE>

sera affiché à l'écran. Si le fichier n'est pas trouvé, le message

NO FILES SELECTED
(Aucun fichier retenu)

s'affichera.

Si le fichier était verrouillé, le message

FILE LOCKED

apparaîtrait, et le fichier ne serait pas supprimé.

Pour revenir au menu, frappez une touche quelconque sauf RETURN, ESC, CTRL, SHIFT ou RESET, lorsque le message

PRESS ANY KEY TO CONTINUE
(Frappez une touche quelconque pour poursuivre)

apparaît.

SPACE ON DISKETTE (Place sur la disquette)

Cette commande vous permet de savoir combien de secteurs sont utilisés, et combien restent libres, sur la disquette par défaut. Pour l'utiliser, tapez 3, et ces informations seront affichées.

QUIT (Quitter)

Cette commande vous permet de quitter normalement le programme. Lorsque vous taperez 9, vous reviendrez dans la situation où vous étiez avant d'exécuter FID.

Que faire en cas d'erreur?

Si une erreur insurmontable est détectée par le programme, un message d'erreur est affiché et l'opération en cours est interrompue. Si une opération de copie est interrompue, il convient d'examiner le catalogue de la disquette copiée (réceptrice), et de supprimer le fichier altéré concerné, s'il existe.

Il porte soit le nom de fichier original, soit un autre nom si vous lui en avez affecté un nouveau, mais il occupe sur disquette une place plus petite que le fichier d'origine, car il est incomplet.

Des messages d'erreur sont engendrés dans les cas de :
manque de place sur la disquette (DISK FULL);
disquette en protection écriture (DISK WRITE PROTECTED);
fichier verrouillé (FILE LOCKED);
erreur d'entrée/sortie, c'est-à-dire de lecture ou d'écriture (I/O ERROR).

Si l'une de ces erreurs est détectée, le programme FID renverra l'exécution au menu principal près que vous ayez frappé une touche quelconque.

Si une erreur DOS est détectée, autre que celles énumérées ci-dessus, le numéro de son code d'erreur sera affiché à l'écran. Ceci se produira si vous avez perdu le DOS, ou si vous avez perdu le programme FID, ou si vous avez tenté de copier un "mauvais fichier" (BAD FILE). Dans tous ces cas, vous devrez à nouveau appeler le DOS, et relancer l'exécution de FID par BRUN FID.

Si, à la suite d'une détection d'erreur, un fichier altéré est écrit sur votre disquette réceptrice, supprimez-le immédiatement, avant de copier un autre fichier sur cette disquette.

ANNEXE K
L'utilisation du programme MUFFIN

180 Le démarrage
183 Les caractères génériques

Le DOS courant est un DOS « 16 secteurs par piste ». Les versions antérieures au 3.3 sont des DOS « 13 secteurs par piste ». Pour plus de détails à propos des secteurs, voyez la section intitulée « GENERALITES SUR LA MEMORISATION » dans l'Annexe C.

Un DOS 16 secteurs ne peut pas fonctionner avec des disquettes qui ont été INITialisées avec un DOS 13 secteurs. Ceci est dû au fait que les informations ne sont pas mémorisées de la même façon sur une disquette 13 secteurs avec un DOS 16 secteurs, le message

UNABLE TO READ/WRITE
(Impossible de lire/écrire)

sera affiché.

Le démarrage

Le programme MUFFIN transforme une disquette 13 secteurs et les programmes qu'elle contient en une disquette 16 secteurs. Ceci est fait en réorganisant les informations qui y sont mémorisées, et en les recopiant sur une autre disquette. La disquette réceptrice, sur laquelle les informations sont écrites doit avoir au préalable été INITialisée avec un DOS 16 secteurs. Après la conversion de 13 secteurs à 16 secteurs, la disquette d'origine (émétrice) comportera toujours la version 13 secteurs.

Pour utiliser au mieux la place disponible sous le DOS 16 secteurs, il vaut mieux convertir tous vos fichiers 13 secteurs au DOS 16 secteurs. Vous pourrez ensuite INITialiser à nouveau vos disquettes 13 secteurs avec le DOS actuel.

Le programme MUFFIN peut fonctionner soit avec un lecteur de disquettes, soit avec deux lecteurs.

L'exemple ci-dessous vous montre comment utiliser MUFFIN pour convertir une disquette 13 secteurs à une disquette 16 secteurs, avec un APPLE comportant un seul lecteur. Il part de l'hypothèse que le lecteur est connecté à la carte d'interface insérée dans le port numéro 6.

1. Insérez la disquette Maître dans votre lecteur, et tapez

   BRUN MUFFIN

2. Lorsque le message

```
*****************************************
*     APPLE II DOS 3.2 TO 3.3 CONVERTER  *
*                                        *
*          MUFFIN VERSION D              *
*                                        *
*     COPYRIGHT 1979 APPLE COMPUTER INC  *
*****************************************
```

CHOOSE ONE OF THE FOLLOWING OPTIONS

<1> CONVERT FILES
<2> QUIT
WHICH WOULD YOU LIKE ?

apparaîtra à l'écran, tapez 1 pour indiquer que vous désirez convertir des fichiers.

3 Vous verrez apparaître la question

SOURCE SLOT?
(Numéro de port de l'original - émetteur?)

Tapez 6 (c est le numéro de port dans lequel est inséré la carte d'interface reliée au lecteur).

A la question

DRIVE?
(Lecteur?)

répondez par 1.

4. Vous verrez apparaître la question

DESTINATION SLOT
(Numéro de port de la copie — récepteur)

Tapez 6, puis à la question DRIVE?, tapez 1.

5 C'est le nom du fichier à convertir qui vous sera ensuite demandé :

FILE NAME?
(Nom du fichier?)

Tapez le signe égal = . Le signe égal est un symbole qui représente le nom de tous les fichiers contenus sur la disquette. Cette réponse signifie donc que vous désirez convertir la totalité de la disquette d'origine (émettrice).

6. Avant que la conversion ne débute, le message

DO YOU WANT PROMPTING?
(Désirez-vous être consulté?)

sera affiché. Pour le moment, tapez N pour non.

7. Lorsque vous verrez le message

INSERT DISKS THEN PRESS <ESC> TO RETURN TO
MAIN MENU OR ANY OTHER KEY TO BEGIN
(Insérez les disquettes puis frappez <ESC> pour revenir au menu principal ou une touche quelconque pour poursuivre).

Retirez la disquette Maître du lecteur, et insérez une disquette 13 secteurs que vous désirez convertir en 16 secteurs. Puis frappez la touche RETURN.

8. Lorsque le message

INSERT SOURCE DISK AND PRESS A KEY
(Insérez la disquette d'origine — émettrice — et frappez une touche)

apparaît à l'écran, il suffit de frapper la touche RETURN.

9 Le programme recherchera maintenant le premier fichier sur votre disquette 13 secteurs, et en affichera le nom. Puis il restera en attente jusqu'à ce que vous remplaciez la disquette d'origine par une disquette initialisée à 16 secteurs. Lorsque vous l'aurez fait, frappez la touche RETURN.

Vous verrez apparaître le message

DONE

lorsque la conversion du fichier est terminée. Le système vous engagera ensuite à remettre la disquette d'origine dans le lecteur. Il convient de répéter cette procédure autant de fois que vous avez de fichiers 13 secteurs à convertir.

Remarque : pour la conversion en utilisant plus de un lecteur de disquettes, il convient de spécifier les numéros de port et de lecteur pour l'original (émetteur) et pour la copie (récepteur) lorsque le système le demande (paragraphe 3 et 4 ci-dessus). Placer les disquettes dans les lecteurs appropriés lorsque le système le demande (paragraphe 8 ci-dessus).

Lorsque le message

INSERT DISKS THE PRESS <ESC> TO RETURN TO
MAIN MENU OR ANY OTHER KEY TO BEGIN
(Insérez les disquettes puis frappez <ESC> pour revenir au menu principal ou une touche quelconque pour poursuivre)

apparaît, il vous est encore possible de changer d'avis : si vous frappez la touche ESC, l'exécution s'interrompra, et reprendra au menu.

Si vous tentez de convertir un fichier 13 secteurs portant le même nom qu'un fichier qui est déjà mémorisé sur la disquette copie (récepteur), le message suivant sera affiché :

FILE (FILENAME)
ALREADY EXISTS
TYPE IN A NEW FILE NAME FOR THE COPY
OR <RETURN> TO REPLACE EXISTING FILE
OR <CTRL-C><RETURN> TO CANCEL COPY
(Le fichier (nom de fichier)

existe déjà

Donnez un nouveau nom pour la copie

ou <RETURN> pour « écraser » le fichier existant
ou <CTRL-C><RETURN> pour annuler la copie)

Vous pouvez choisir de :

– donner un nouveau nom de fichier à convertir :
– convertir le fichier 13 secteurs et « écraser » le fichier 16 secteurs qui porte le même nom sur la disquette copie (réceptrice), en tapant RETURN ;
– interrompre la conversion en tapant CTRL-C RETURN.

L'utilisation des caractères génériques

Comme vous avez pu le constater dans l'exemple ci-dessus, le signe égal = est utilisé pour signifier «tous les fichiers sur la disquette». Le signe égal peut aussi être donné pour un quelconque caractère, ou groupe de caractères à l'intérieur d'un nom de fichier. Par exemple, à la question

FILENAME?
(Nom de fichier?)

vous répondez en tapant

FI = LE

vous convertirez en 16 secteurs tous les fichiers dont le nom débute par FI et se termine par LE. Suivant le même principe, la réponse

= TEXT

aurait pour effet la conversion de tous les fichiers dont le nom se termine par TEXT. De même la réponse =*= aurait pour effet la conversion de tous les fichiers dont le nom comporte un astérisque *.

La question

DO YOU WANT PROMPTING?
(Désirez-vous être consulté?)

apparaîtra si vous utilisez le caractère générique. Si vous répondez Y (yes - oui) à cette question, le programme s'interrompra après avoir repéré chaque fichier sur la disquette d'origine à 13 secteurs, et attendra vos instructions (Y pour convertir, N pour ne pas convertir, Q - quit - pour interrompre).

Si vous répondez N (no - non) à la question

DO YOU WANT PROMPTING?
(Désirez-vous être consulté?)

tous les fichiers compatibles avec le caractère générique donné seront convertis.

Index

186 Index général
189 Index des ménages
189 Index des programmes
Fiches cartonnées : Index des résumés de commandes
Index des résumés des procédures DOS

Index général

A

Appel de DOS 11-12, 156
  13 secteurs 170
APPEND 64-65, 150
Applesoft BASIC 28-29

B

BASIC simple 28-29
BASIC à virgule flottante : voir BASIC étendu (Applesoft)
BLOAD 89, 154
Bloc de gestion E/S (de contrôle) 90-93
  (Input/output control block = IOB)
BRUN 89, 89, 154
BSAVE 88, 153

C

Câble ruban 2-4
CALL –142 29
CALL –868 43
Caractère d'appel 11, 36
Caractères génériques 172-181
  (wildcard character)
Carte interface 2, 4
  (controller card)
Carte ROM firmware Applesoft 101
Cassette, enregistreur 15, 25
Catalogue 16, 144
CHAIN 100, 149
Champ d'adresses 90
Champ de données 86
CHR$ 30
CLOSE 149, 16
Code d'erreur 108-109
Connecteur 3
  (slot)
CONTACT 2
Copie de disquettes 38-40
Copie de programmes 15-16
Copie de fichier texte 127

C

CTRL 11, 140
CTRL-C 18, 39
CTRL-D 30-31, 147-148, 156
CTRL-K 99
CTRL-P 99

D

Déballage 2
Debugging : voir mise au point
DELETE 18, 26, 145
Disk(ette)
  BASIC 170
  CATALOG 16, 144
  manipulation de, 5-6
  répertoire de format secteur 122-124
Disquette « maître » 44-46
Disquette « maître » de système 10-13-14
Disquette « esclave » 13, 133
D$ 30
DOS (Disk Operating System)
  résumé de commandes du, 104-105, 140-155
  commande mode indirect du, 31, 156
    (de l'intérieur d'un programme)
  registres E/S du, 97, 100
  utilisation des mémoires du, 132
  messages du DOS 108, 116
  résumé des procédures du, 156-160

E

Effacement des fichiers 18, 26, 145
Enchaînement de l'Applesoft 100, 160
END OF DATA †11
Enregistrement des mémoires 78, 120
ERREUR E/S (I/O ERROR) 113
ESC (escape) 11, 141
EXEC, fichier 71

F

FID 172-177
Fichier 16
  nom de, 16-17, 25, 142
Fichiers binaires 88
Fichiers de données : voir fichier texte
Fichier EXEC 71, 159
  création de, 71

Fichier langage machine 89
Fichier texte 48
Fichier de texte accès aléatoire 79, 84
création et extraction de, 78, 81, 159
différence de ce fichier avec le fichier à accès séquentiel 78
programmes échantillons de, 78, 83
Fichier de texte à accès séquentiel 49-68
création et extraction de, 49-68, 157-158
programme échantillon de, 49, 68
FILE NOT FOUND 112
FILE LOCKED 114
FILE TYPE MISMATCH 115
Format 90
FP 28-29

G

GET 51
Guillemets 30

H

HIMEM 12, 133

I

IN# 11, 96, 98, 148
INIT (Initialise) 13, 31, 142, 156
Initialisation 13-14, 18, 156
INPUT 51
Insertion et retrait 6-7
Installation du lecteur DISK II 2-4
INT 28-29, 147

J

JMP (jump) 89

K

Kick (ou boot) : voir amorcer

L

L (longueur), paramètre
d'un fichier binaire 88
d'un enregistrement 84, 120
LANGUAGE NOT AVAILABLE 109
avec BSAVE 88, 141
avec OPEN 84, 120, 141
Lecteur de disquettes
dépannage 12
DISK FULL 144
Emploi 5
Installation 2-4
Installation de plusieurs lecteurs de disquettes 5, 22
Limites des mémoires 13, 132
Lire ou écrire une piste
ou un secteur 90, 93
Liste piste/secteur 118, 121-122
LOAD 16, 25-26, 144
LOCK 35, 145

M

MAKE TEXT 60-62
Manuel de référence de programme Applesoft BASIC II (10, 48)
Manuel de référence de programmation BASIC Apple II 10
MASTER CREATE 44-46
MAXFILES 77, 43-44, 146-147
Mémoires-tampons de fichier 43, 118
Messages d'erreur 108-116
voir aussi INDEX des MESSAGES
Mise au point (42)
(debugging)
Mode exécution différée 29, 31, 48
Mode exécution immédiate 31, 48
Mode moniteur 50
MON 42-43, 146

N

NO BUFFERS AVAILABLE 125
NOMON 42-43, 146
NOT DIRECT COMMAND 116
Notation 140-141
Notation hexadécimale 24
Numéro de volume 23

O

ONERR GOTO 108, 114
OPEN 49-68, 149, 152
Option d'affichage : voir MON, NOMON
Option lecteur : voir paramètre lecteur
Ordre d'affectation pistes 128

P

PR# 11, 96, 98, 148
Paramètre (d'adresse) A 88-141
avec BLOAD 89
avec BRUN 89
Paramètre de champ (R)
avec EXEC 75
avec POSITION 65-67
Paramètre (de commande)
avec MON (42)
avec NOMON 42
Paramètre connecteur 22, 141
(slot 5)
Paramètre d'entrée (I) 42
Paramètre de lecteur 22
Paramètre de nº d'enregistrement 78, 82
Paramètre octet absolu 67-68, 119
Paramètre octet B 67, 69
avec READ 67-68, 84, 141
avec WRITE 67-68, 84, 141
de copie de disquette 37-37
Paramètre R
avec EXEC 75
avec POSITION 65-67, 119
avec READ 65-66, 141
avec WRITE 65-66, 141
Paramètre de sortie
avec MON 42
avec NOMON 42
Piste 90, 118
POSITION 64-66, 119, 150-151
Position champ relatif, paramètre R 65-67
Position champ absolu
paramètre de, 75
PRINT (avec CTRL-D) 29-31, 50
PROGRAM TOO LARGE 29
Programme Applesoft 29
Programme COPY 38-40
Programme COPYA 38-40
Programme HELLO 13
Programme MUFFIN 180-182
Programme salutations 13-14
changement de nom de, 45
PROMS 162-164
Protection d'écriture 36-37

R

RANGE ERROR 110
READ
avec fichier texte accès aléatoire 84-85
avec fichier texte accès séquentiel 68, 129, 150
Registres E/S 97, 100
Registres DOS 97, 99
Registres Moniteur 97, 99
Registre Y 90-91
RENAME 17, 144
RESET 11, 18, 141, 156
RETRIEVE TEXT 63-64
RETURN 11, 12, 141
RETURN, caractère 51
Routine RWTS 90-93
RUN 25-26, 144

S

SAVE 15, 25-26, 144
Schéma 137-138
Schéma circuit interface 137
Schéma des pistes 118, 126-127
(track bit map)
Schéma unité analogique 138
Secteur 16, 90, 118
ordre affectation des, 128
conversion de 13 en 6 180-182
format 121
Stockage 118
Surimpression 61-62, 67
Syntaxe 140-141
SYNTAX ERROR 114
Système clés en main 34-35, 157, 165

T

Table de caractéristique d'unité 90-93
Texte, fichier, aléatoire : voir fichier
Texte, fichier, accès séquentiel : voir fichier
TRACE 44

U

Unités E/S 96
UNLOCK 35, 145
Utilisation des mémoires et cartes mémoires 132

V

Valeurs par défaut 22
VERIFY 35-36, 145
VOLUME MISMATCH 112-113
Valeur, numéro : voir paramètre de volume
Valeur (V), paramètre 23, 141
Voyant « IN USE » 7, 18
VTOC (valeur de la table des matières) 125-126

W

WRITE
avec fichier texte accès aléatoire 84-85, 120, 153
avec fichier texte accès séquentiel 49-68, 118-119, 150
WRITE PROTECTED 110

X

XFER : voir commande XF

Index des messages

PROGRAM

Salutations (SALUT) 13-14
COUNT 15
CATALOG-Greeting 30–31
COLOR DEMO 34-35
ANIMALS 37
COPY 38-40
MASTER CREATE 44-46
EXEC DEMO 70-71
CAPTURE 72-73
RANDOM, APPLE PROMS 82-83
CHAIN 100-101
FID 172-177
MUFFIN 180-182

Index des programmes

ERROR #	MESSAGE
9	DISK FULL 114
5	END OF DATA 53, 58, 65, 66, 67, 75, 82, 111-112, 151, 152
10	FILE LOCKED 35, 65, 114
6	FILE NOT FOUND 17, 18, 26, 36, 46, 59, 112, 145
13	FILE TYPE MISMATCH 35, 48, 115, 144
8	I/O ERROR 22, 26, 36, 113, 145
1	LANGUAGE NOT AVAILABLE 28, 109, 144
12	NO BUFFERS AVAILABLE 43, 115, 147
15	NOT DIRECT COMMAND 48, 116
14	PROGRAM TOO LARGE 29, 116
2, 3	RANGE ERROR 92, 110
11	SYNTAX ERROR 10, 26, 28, 75, 88, 114
7	VOLUME MISMATCH 23, 112-113
4	WRITE PROTECTED 37, 110-111

Fiche de référence rapide du DOS

Sur cette carte, les commandes DOS sont groupées suivant les trois catégories suivantes :

Commandes de service :

INIT	LOAD	DELETE	VERIFY	MAXFILES
CATALOG	RUN	LOCK	MON	
SAVE	RENAME	UNLOCK	NOMON	

Commandes d'accès :

FP	INT	PR#	IN#	CHAIN

Commandes fichier texte accès séquentiel :

OPEN	READ	APPEND	EXEC
CLOSE	WRITE	POSITION	

Commandes fichier texte accès aléatoire :

OPEN	CLOSE	READ	WRITE

Commandes fichiers langage machine :

BLOAD	BRUN	BSAVE

Notation et syntaxe

Un paramètre est une lettre capitale normalement suivie par un chiffre (indiqué ici par une minuscule) qui fournit une information supplémentaire pour l'exécution de la commande. Des paramètres multiples peuvent apparaître dans un ordre quelconque mais doivent être séparés les uns des autres par une virgule. Un paramètre apparaissant à l'intérieur de parenthèses droites [comme celles-ci] est optionnel.

Un nom de fichier (indiqué ici par X) doit suivre immédiatement son mot de commande. Les noms de fichier doivent commencer par une lettre; seuls les 30 premiers caractères sont utilisés. Une virgule sépare un nom de fichier du paramètre suivant.

CTRL-D (taper D tout en maintenant la touche CTRL enfoncée) est utilisé dans les expressions PRINT pour indiquer le début d'une commande DOS à exécution différée.

Exemple de BASIC simple :

```
10 D$ ="" : REM "CTRL-D"
20 PRINT D$; "CATALOG"
```

Exemple de BASIC Applesoft :

```
10 D$ = CHR$(4) : REM CTRL-D
20 PRINT D$; "CATALOG"
```

Le terme BASIC seul signifie que le BASIC simple ou le BASIC Applesoft sont utilisés. Le terme « fichier » seul signifie tout type de fichier sur disquette.

Paramètres de commande

Un message d'erreur est issu lorsqu'une « quantité » de commande DOS est trop grande ou trop petite.

TOUS FICHIERS

Paramètres	Tel indiqué	Min.	Max.
Connect.	,Ss	S1	S7
Lecteur	,Dd	D1	D2
Volume	,Vv	VØ*	V254

* L'utilisation de VØ est similaire à l'omission du paramètre Vv : le numéro de volume de la disquette est ignoré. Le plus petit numéro de volume que INIT pourra en fait affecter à une disquette est 1.

FICHIER DE TEXTE A ACCÈS SÉQUENTIEL

Paramètre	Tel indiqué	Min.	Max.
Octet	,Bb	BØ	B32767
Champ relatif*	,Rp	RØ	R32767

* Avec EXEC, toujours relatif à champ Ø.

FICHIERS TEXTE A ACCÈS ALÉATOIRE

Paramètre	Tel indiqué	Min.	Max.
Longueur enregistrement	,Lj	L1	L32767
Numéro d'enregistrement	,Rr	RØ	R32767

FICHIERS BINAIRES

Paramètre	Tel indiqué	Min.	Max.
Adresse de début	,An	AØ	A65535
Nombre d'octets	,Lj	L1	A32767

COMMANDES DOS

Commande	Quantité	Tel indiqué	Min.	Max.
PR#	conn. (slot)	PR# s	PR#Ø	PR#7
IN#	conn. (slot)	IN# s	IN#Ø	IN#7
MAXFILES	mémoire-tamp.	fichiers MAXFILES n	n=1	N=16

Les commandes utilisent les paramètres de connec. (slot) ou de lecteur (drive) lorsqu'elles passent sur un connecteur (slot) ou lecteur (drive) différent.

Si une commande omet le paramètre Volume ou utilise VØ, le numéro de volume de disquette est ignoré. Une commande qui utilise le paramètre de Volume Vv ne sera exécutée que si le numéro de volume de disquette est v.

Commandes de service

INIT X [,Vv] [,Ss] [,Dd]
Initialise une disquette « vierge » pour former une disquette esclave. Affecte le nom de programme de salutations X et le numéro de volume v (s'il est spécifié). Sauvegarde (SAVE) le programme BASIC normalement en mémoire sous le nom de fichier X.

CATALOG [,Ss] [,Dd]
Affiche le numéro de volume et tous les fichiers sur une disquette avec chacun des types de fichiers et la longueur de secteur. * indique un fichier verrouillé (LOCKed).

Type	Description	Comment a été créé
I	Fichier de programme BASIC simple	(SAVE)
A	Fichier de programme BASIC Applesoft	(SAVE)
T	Fichier de texte	(OPEN, puis WRITE)
B	Fichier à image-mémoire binaire	(BSAVE)

SAVE X [,Ss] [,Dd] [,Vv]
Stocke le programme BASIC courant sur la disquette, sous le nom de fichier X. Surimprime tout fichier antérieur de même type et nom, sans avertissement.

LOAD X [,Ss] [,Dd] [,Vv]
Charge le fichier de programme BASIC (X) dans la mémoire après avoir vidé la mémoire et (si nécessaire) modifié sur le BASIC approprié.

RUN X [,Ss] [,Dd] [,Vv]
Charge (LOAD) le fichier X programme BASIC puis déroule (RUN) le programme.

RENAME X, Y [,Ss] [,Dd] [,Vv]
Modifie le nom d'un fichier de disquette de X en Y.

DELETE X [,Ss] [,Dd] [,Vv]
Efface le fichier X de la disquette.

LOCK X [,Ss] [,Dd] [,Vv]
Verrouille le fichier X contre toute modification, fichier verrouillé (LOCKed) indiqué dans CATALOG par *.

UNLOCK X [,Ss] [,Dd] [,Vv]
Libère le fichier verrouillé (LOCKed) pour autoriser une modification ou une suppression.

VERIFY X [,Ss] [,Dd] [,Vv]
Vérifie la consistance interne du fichier X. Si X a été sauvegardé sans erreur, aucun message n'est donné.

MON [,C] [,I] [,O]
Déclenche l'affichage des Commandes de lecteur (C), l'entrée (Input) du lecteur (I) et la sortie (Output) vers le lecteur (O). Sans paramètres, MON est ignorée.

NOMON [,C] [,I] [,O]
Supprime l'affichage des Commandes de lecteur (C), l'entrée (Input) à partir du lecteur (I) et la sortie (Output) vers le lecteur (O). Sans paramètres, NOMON est ignorée.

MAXFILES n
Réserve un nombre n de mémoires-tampons de fichier pour l'entrée et la sortie du lecteur (l'amorce réserve 3 mémoires-tampons de fichier). Utiliser avant le chargement (LOADing) et le déroulement (RUNning) d'un programme.

Commandes d'accès

FP [,Ss] [,Dd] [,Vv]
Place le système dans le BASIC Applesoft, en effaçant tout programme en mémoire.

INT
Place le système dans le BASIC simple, en effaçant tout programme en mémoire.

PR# s
Envoie la sortie ultérieure sur le connecteur (slot) s. Amorce le lecteur si le connecteur s contient une carte d'interface. PR#0 envoie la sortie de nouveau sur l'écran TV.

IN# s
Prend une entrée ultérieure à partir du connecteur s. Amorce le lecteur si le connecteur s contient une carte d'interface. IN#0 récupère de nouveau la sortie à partir du clavier.

CHAIN Y [,Ss] [,Dd] [,Vv]
Déroule (RUN) le fichier Y de programme BASIC simple, mais ne retire pas les variables développées par un programme BASIC simple antérieur.

Commandes de fichier de texte à accès séquentiel

OPEN X [,Ss] [,Dd] [,Vv]
Ouvre ou crée un fichier X de texte à accès séquentiel. Affecte une mémoire-tampon de fichier et prépare l'écriture (WRITE) ou la lecture (READ) à partir du début du fichier.

CLOSE [X]
Complète WRITE X, si nécessaire, et désaffecte la mémoire-tampon de fichier affectée du fichier de texte X. Sans nom de fichier, ferme (CLOSE) tous les fichiers ouverts (OPEN) (à l'exception du fichier EXEC).

WRITE X [,Bb]
Des PRINT ultérieurs envoient des caractères au fichier de texte séquentiel X. L'écriture (WRITE) commence sur la position courante de fichier ou (si spécifié) sur l'octet b. Supprimé par toute commande DOS.

READ X [,Bb]
Des INPUT et des GET prennent les caractères de réponse à partir du fichier X de texte à accès séquentiel. La lecture (READing) commence sur la position courante de fichier ou (si spécifié) sur l'octet b. La réponse INPUT est à un champ (tous les caractères sur le RETURN suivant). Supprimé par toute commande DOS.

APPEND X [,Se] [,Dd] [,Vv]
Ouvre le fichier X de texte (séquentiel) existant, identique à OPEN mais prépare l'écriture (WRITE) à la fin du fichier.

POSITION X, Rp
Dans un fichier ouvert texte séquentiel X, un READ ou WRITE ultérieurs reprendront à partir du champ p-th à la suite de la position courante de fichier.

EXEC X [,Rp] [,Ss] [,Dd] [,Vv]
Exécute des champs successifs dans un fichier texte séquentiel X comme s'ils ont été tapés sur le clavier. Avec le paramètre Rp, l'exécution commence sur le champ p-th. Les champs peuvent comprendre des lignes de programme BASIC numérotées et des commandes BASIC à exécution directe ou DOS pour gérer l'Apple.

Commandes de fichier de texte à accès aléatoire

OPEN X, Lj [,Ss] [,Dd] [,Vv]
Ouvre ou crée un fichier X de texte à accès aléatoire, affecte une mémoire-tampon de fichier et définit la longueur d'un enregistrement suivant j octets; prépare la lecture (READ) et l'écriture (WRITE) à partir du début de l'enregistrement 0. Un paramètre de même longueur doit être utilisé chaque fois que le fichier X est ouvert (OPENed).

CLOSE X [,Ss] [,Dd] [,Vv]
Complète WRITE X, si nécessaire et désaffecte la mémoire-tampon de fichier affectée au fichier de texte X. Sans nom de fichier, ferme (CLOSE) tous les fichiers ouverts (OPEN).

WRITE X [,Rr] [,Bb]
Des PRINTS ultérieurs envoient des caractères au fichier X de texte à accès aléatoire. Sans paramètres, l'écriture (WRITEing) commence sur la position courante de fichier. Avec le paramètre Rr seul, l'écriture (WRITEing) commence sur l'octet 0 de l'enregistrement r. Avec le paramètre Bb, l'écriture WRITE commence sur l'octet b de l'enregistrement spécifié. Supprimé par toute commande DOS.

READ X [,Rr] [,Bb]
Des INPUT et des GET ultérieurs prennent les caractères de réponse à partir du fichier X de texte accès aléatoire. Sans paramètres, la lecture (READing) commence sur la position courante de fichier. Avec le paramètre Rr seul, la lecture (READing) commence sur l'octet 0 de l'enregistrement r. Avec le paramètre Bb, la lecture (READing) commence sur l'octet b de l'enregistrement courant ou spécifié. La réponse INPUT est à un champ (tous les caractères sur le RETURN suivant). Supprimé par toute commande DOS.

Commande de fichier langage machine

BSAV X, Aa, Lj [,Ss] [,Dd] [,Vv]
Stocke sur disquette, sous le nom de fichier X, le contenu des octets de mémoire j à partir de l'adresse a.

BLOAD X [,Aa] [,Ss] [,Dd] [,Vv]
Charge le fichier binaire X dans les mêmes positions de mémoire à partir desquelles le fichier a été sauvegardé (BSAVEd) ou (si spécifié) à partir de l'adresse a.

BRUN X [,Aa] [,Ss] [,Dd] [,Vv]
Charge (BLOAD) le fichier binaire X puis saute (JMP) sur la première adresse mémoire de fichier chargée.

apple computer

Printed in Ireland. A2LF0036. 030-F115-00.